**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**





**TIỂU LUẬN KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG**

[**DEMAND FORECASTING & INVENTORY OPTIMIZATION**](https://thecleverprogrammer.com/2023/08/28/demand-forecasting-and-inventory-optimization-using-python/)

**DỰ BÁO NHU CẦU & TỐI ƯU HOÁ HÀNG TỒN KHO**

**Giảng viên hướng dẫn: TS. Võ Thị Hồng Thắm**

**Sinh viên thực hiện:**

**Huỳnh Quốc Khánh - 2100007626**

**Trần Đình Hùng - 2100004753**

**Trần Hồng Kính - 2100007981**

**Nguyễn Quốc Tấn Lộc - 2100004516**

**Khoá: 2021**

**Ngành/ chuyên ngành: Khoa học dữ liệu**

Tp HCM, 27 tháng 05 năm 2024

# LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin Trường Đại Học Nguyễn Tất Thành đã truyền đạt vốn kiến thức quý báu của quý thầy cô trong suốt thời gian học tập tại trường.Nhờ kiến thức đã truyền đạt và sự dẫn dắt của cô mà đề tài [Demand forecasting & inventory optimization](https://thecleverprogrammer.com/2023/08/28/demand-forecasting-and-inventory-optimization-using-python/) được hoàn thành theo dự kiến

Em xin chân thành cảm ơn đến cô TS.Võ Thị Hồng Thắm –đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này

Bài báo của em còn nhiều hạn chế mong cô xem xét và đưa ra nhận xét để em có thể khắc phục và nâng cao trình độ của mình trong chuyên ngành hơn nữa là để phát triển bản thân

Em xin chân thành cảm cô rất nhiều !

Sinh viên thực hiện

Huỳnh Quốc Khánh

# LỜI MỞ ĐẦU

Nhu cầu và tối ưu hàng hóa tồn kho là hai khía cạnh quan trọng của quản lý chuỗi cung ứng vì chúng đều có vai trò quan trọng trong việc đảm bảo hoạt động kinh doanh được hiệu quả và linh hoạt

Dự báo nhu cầu có nghĩa là ước tính nhu cầu của khách hàng trong tương lai đối với một sản phẩm hoặc dịch vụ dựa trên dữ liệu lịch sử và các yếu tố liên quan. Tối ưu hoá hàng tồn kho là quản lý chiến lược mức tồn kho để đảm bảo có đủ lượng hàng hoá vào đúng thời điểm nhằm đáp ứng nhu cầu của khách hàng đồng thời giảm thiểu chi phí

Dự báo nhu cầu chính xác cho phép các công ty đáp ứng nhu cầu của khách hàng, tránh tình trạng tồn kho quá mức hoặc thiếu hàng và tối ưu hoá hoạt động chuỗi cung ứng của họ.

Tối ưu hoá hàng tồn kho hiệu quả giúp doanh nghiệp giảm chi phí vận chuyển, cải thiện dòng tiền và nâng cao sự hài lòng của khách hàng.

Ở tiểu luận này,chúng em sử dụng mô hình dự báo chuỗi thời gian Prophet để dự báo nhu cầu và mô hình Random Forest để xem xét các đặc trưng ảnh hưởng đến doanh thu .Sau đó, chúng em sử dụng kết quả dự báo được để tính toán tối ưu hoá mức tồn kho và kiểm soát hàng bị thiếu hụt bằng cách thực hiện tính toán Reorder Point (điểm đặt hàng lại),kho an toàn và Economic Order Quantity (EOQ - Số lượng đặt hàng kinh tế)

Chúng ta hãy cùng nhau đi tìm hiểu nào !

| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ II NĂM HỌC 2024 -2025** |
| --- | --- |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

Môn thi: Khai thác dữ liệu và ứng dụng Lớp học phần: 21DTH3A

Nhóm sinh viên thực hiện :4

1 Huỳnh Quốc Khánh Tham gia đóng góp:100%

2 Trần Đình Hùng . Tham gia đóng góp:100%

3 Trần Hồng Kính Tham gia đóng góp:100%

4 Nguyễn Quốc Tấn Lộc Tham gia đóng góp:100%

Ngày thi: 27/05/2024 Phòngthi:L.712

Đề tài tiểu luận/báo cáo của sinh viên :[Demand forecasting & inventory optimization](https://thecleverprogrammer.com/2023/08/28/demand-forecasting-and-inventory-optimization-using-python/)

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| --- | --- | --- | --- |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  |  |  |
| * Lập luận |  |  |  |
| * Kết luận |  |  |  |
| Trình bày |  |  |  |
| **TỔNG ĐIỂM** |  |  |  |

|  | **Giảng viên chấm thi**  Võ Thị Hồng Thắm |
| --- | --- |

**MỤC LỤC**

[**LỜI CẢM ƠN 1**](#_heading=h.gjdgxs)

[**LỜI MỞ ĐẦU 2**](#_heading=h.30j0zll)

[**DANH MỤC HÌNH 6**](#_heading=h.3znysh7)

[**CHƯƠNG 1 GIỚI THIỆU CHUNG 7**](#_heading=h.tyjcwt)

[*1.1 Tính cấp thiết và lý do chọn đề tài 7*](#_heading=h.3dy6vkm)

[*1.2 Mục đích 7*](#_heading=h.1t3h5sf)

[*1.3 Cách thực hiện 7*](#_heading=h.4d34og8)

[*1.4 Câu hỏi nguyên cứu 7*](#_heading=h.2s8eyo1)

[*1.5 Đối tượng và phạm vi trong nguyên cứu 8*](#_heading=h.17dp8vu)

[*1.6 Ý nghĩa khoa học và thực tiễn của đề tài 8*](#_heading=h.3rdcrjn)

[**CHƯƠNG 2 MÔ HÌNH PROPHET, RANDOM FOREST 9**](#_heading=h.26in1rg)

[**2.1 Mô hình Prophet 9**](#_heading=h.lnxbz9)

[*2.1.1 Lý thuyết 9*](#_heading=h.35nkun2)

[*2.1.2 Hàm tăng trưởng 10*](#_heading=h.1ksv4uv)

[*2.1.3 Hàm thời vụ 10*](#_heading=h.44sinio)

[*2.1.4 Ngày lễ,sự kiện 11*](#_heading=h.3j2qqm3)

[*2.1.5 Ưu điểm,nhược điểm 11*](#_heading=h.1y810tw)

[**2.2 Mô hình Random Forest 12**](#_heading=h.4i7ojhp)

[*2.2.1 Khái niệm 12*](#_heading=h.2xcytpi)

[*2.2.2 Ý tưởng của Random Forest 13*](#_heading=h.1ci93xb)

[*2.2.3 Kết hợp các mô hình (ensemble model) 13*](#_heading=h.3whwml4)

[*2.2.4 Lấy mẫu (boostrapping) 14*](#_heading=h.qsh70q)

[*2.2.5 Đánh giá mô hình bằng mẫu nằm ngoài túi(out of bag) 17*](#_heading=h.1pxezwc)

[*2.2.6 Mô hình rừng cây 17*](#_heading=h.49x2ik5)

[*2.2.7 Huấn luyện mô hình rừng cây 18*](#_heading=h.147n2zr)

[**CHƯƠNG 3 ONE-HOT-ENCODING,LABEL ENCODER 20**](#_heading=h.23ckvvd)

[**3.1 One-Hot-Encoding 20**](#_heading=h.ihv636)

[*3.1.1 Khái niệm 20*](#_heading=h.32hioqz)

[*3.1.2 Quy trình hoạt động 20*](#_heading=h.1hmsyys)

[*3.1.3 Ưu điểm,nhược điểm 21*](#_heading=h.2grqrue)

[**3.2 Label Encoder 21**](#_heading=h.vx1227)

[*3.2.1 Khái niệm 21*](#_heading=h.3fwokq0)

[*3.2.2 Quy trình hoạt động 22*](#_heading=h.4f1mdlm)

[*3.2.3 Ưu điểm,nhược điểm 22*](#_heading=h.2u6wntf)

[\_Toc23387](#_heading=h.19c6y18)[**CHƯƠNG 4 TINH CHỈNH THAM SỐ MÔ HÌNH VÀ LỰA CHỌN ĐẶC TRƯNG 20**](#_heading=h.23ckvvd)**6**

[**4.1 Parameter Grid 23**](#_heading=h.3tbugp1)

[*4.1.1 Khái niệm 23*](#_heading=h.28h4qwu)

[*4.1.2 Mục đích của Parameter Grid 23*](#_heading=h.37m2jsg)

[*4.1.3 Cách sử dụng Parameter Grid 24*](#_heading=h.1mrcu09)

[**4.2 Thuộc tính feature\_importance\_ 24**](#_heading=h.46r0co2)

[*4.2.1 Khái niệm 24*](#_heading=h.2lwamvv)

[*4.2.2 Cách tính feature\_importance\_ 25*](#_heading=h.3l18frh)

[*4.2.3 Mục đích của feature\_importance\_ 25*](#_heading=h.206ipza)

[**CHƯƠNG 5 ĐÁNH GIÁ HIỆU SUẤT MÔ HÌNH 26**](#_heading=h.4k668n3)

[*5.1 .Cross-validation trong prophet 26*](#_heading=h.2zbgiuw)

[*5.2 Song song trong cross\_validation 27*](#_heading=h.1egqt2p)

[*5.3 Performance\_metrics trong prophet 28*](#_heading=h.3ygebqi)

[*5.4 Plot\_cross\_validation\_metric trong prophet 29*](#_heading=h.sqyw64)

[**CHƯƠNG 6 REORDER POINT,SAFETY STOCK,ECONOMIC ORDER QUANTITY - EOQ 30**](#_heading=h.3cqmetx)

[**6.1 Reorder point 30**](#_heading=h.1rvwp1q)

[*6.1.1 Khái niệm 30*](#_heading=h.4bvk7pj)

[*6.1.2 Tầm quan trọng cảu điểm đặt hàng lại 30*](#_heading=h.2r0uhxc)

[**6.2 Safety stock 31**](#_heading=h.1664s55)

[**6.3 Economic order quantity - EOQ 31**](#_heading=h.3q5sasy)

[**CHƯƠNG 7 XÂY DỰNG MÔ HÌNH 32**](#_heading=h.25b2l0r)

[**7.1 Giới thiệu bài toán 32**](#_heading=h.kgcv8k)

[*7.1.1. Giới thiệu data 32*](#_heading=h.34g0dwd)

[*7.1.2. Bài toán đặt ra 32*](#_heading=h.1jlao46)

[*7.1.3. Giải quyết vấn đề 32*](#_heading=h.43ky6rz)

[**7.2 Tìm hiểu dữ liệu và xây dựng mô hình 33**](#_heading=h.2iq8gzs)

[*7.2.1 Tìm hiểu dữ liệu 33*](#_heading=h.xvir7l)

[*7.2.2 Tiền xử lý dữ liệu 34*](#_heading=h.3hv69ve)

[*7.2.3 Trực quan hóa 38*](#_heading=h.1x0gk37)

[*7.2.4 Xây dựng mô hình 40*](#_heading=h.4h042r0)

[*7.2.5 Tinh chỉnh mô hình 42*](#_heading=h.2w5ecyt)

[*7.2.6 Tính toán mức đặt hàng lại,mức tồn kho an toàn,mức đặt hàng 46*](#_heading=h.1baon6m)

[*7.2.7 Các yếu tố ảnh hưởng đến kết quả dự đoán 48*](#_heading=h.3vac5uf)

[**CHƯƠNG 8 TỔNG KẾT 51**](#_heading=h.2afmg28)

[**8.1 Tóm tắt nội dung và kết luận 51**](#_heading=h.pkwqa1)

[**8.2 Kết quả đạt được và điểm còn hạn chế 52**](#_heading=h.39kk8xu)

[**DANH MỤC TÀI LIỆU THAM KHẢO 53**](#_heading=h.48pi1tg)

**DANH MỤC HÌNH**

Hình 2.1 Công thức chuỗi Fourier........................................................................[1](#_heading=h.3as4poj)0

Hình 2.2 Mô hình Random Forest Classifier........................................................[1](#_heading=h.3as4poj)2

Hình 2.3 Mô hình kết hợp.....................................................................................[1](#_heading=h.3as4poj)3

Hình 2.4 Lấy mẫu tái lập.......................................................................................[1](#_heading=h.3as4poj)5

Hình 2.5 Sơ đồ mô hình cây quyết định................................................................18

Hình 2.6 Các tham số mô hình..............................................................................[1](#_heading=h.3as4poj)9

Hình 3.1 Ví dụ minh hoạ mã hoá one-hot .............................................................20

Hình 3.2 Ví dụ minh hoạ mã hoá Label.................................................................[2](#_heading=h.1v1yuxt)2

Hình 4.1 Ví dụ sử dụng thuộc tính feature\_importance\_.......................................[2](#_heading=h.111kx3o)5

Hình 5.1 Đầu ra của cross\_validation....................................................................28

Hình 5.2 Kết quả của performance\_metrics...........................................................[3](#_heading=h.2dlolyb)0

Hình 5.3 Kết quả plot\_cross\_validation\_metric.....................................................30

# GIỚI THIỆU CHUNG

## Tính cấp thiết và lý do chọn đề tài

Trong môi trường cạnh tranh gay gắt hiện nay, các doanh nghiệp cần phải tối ưu hoá hiệu quả hoạt động của mình để có thể cạnh tranh hiệu quả. Dự báo nhu cầu và tối ưu hoá kho hàng là những công cụ quan trọng giúp các doanh nghiệp giảm thiểu chi phí, nâng cao hiệu quả và tăng cường lợi thế cạnh tranh.

Demand Forecasting & Inventory Optimization có thể được áp dụng cho nhiều ngành nghề và lĩnh vực khác nhau, từ sản xuất, bán lẻ đến dịch vụ. Do đó, đây là một đề tài có tiềm năng ứng dụng cao và có thể mang lại nhiều lợi ích cho các doanh nghiệp.

## 1.2 Mục đích

Đưa ra các kế hoạch,giải pháp về vấn đề đặt hàng và kho bãi giúp doanh nghiệp giảm thiểu chi phí,nâng cao hiệu quả hoạt động và đặc biệt là đáp ứng được nhu cầu của khách hàng kịp thời nhằm nâng cao lợi thế cạnh tranh trong kinh doanh

## 1.3 Cách thực hiện

B1:Lựa chọn trích xuất dữ liệu

B2:Tiền xử lý dữ liệu

B3:Áp dụng thuật toán

B4:Đưa ra kết quả

## 1.4 Câu hỏi nguyên cứu

Doanh thu của sản phẩm bán chạy nhất trong 12 tháng tới là bao nhiêu?

Mức tồn kho an toàn,mức đặt hàng lại và số lượng đặt hàng cho mỗi lần đặt lại cho sản phẩm bán chạy nhất là bao nhiêu?

## 1.5 Đối tượng và phạm vi trong nguyên cứu

- Đối tượng nghiên cứu:

Nhu cầu của sản phầm bán chạy nhất của doanh nghiệp trong 12 tháng tới

Mức tồn kho an toàn và mức đặt hàng lại và số lượng đặt hàng lại cho sản phẩm bán chạy nhất

- Phạm vi:

Dữ liệu bán hàng của doanh nghiệp ở vài năm trước đến hiện tại

## 1.6 Ý nghĩa khoa học và thực tiễn của đề tài

- Ý nghĩa khoa học:

Đóng góp vào kho tàn tri thức về dự báo nhu cầu và tối ưu hoá kho hàng

Phát triển ứng dụng của trí tuệ nhân tạo và học máy trong quản lý chuỗi cung ứng

Thúc đẩy đổi mới và phát triển trong lĩnh vực quản lý chuỗi cung ứng

- Ý nghĩa thực tiễn:

Giúp doanh nghiệp tối ưu hoá hoạt đông quản lý hàng tồn kho

Nâng cao hiệu quả hoạt động của chuỗi cung ứng

Thích nghi với biến động thị trường

# MÔ HÌNH PROPHET, RANDOM FOREST

## Mô hình Prophet

### 2.1.1 Lý thuyết

Prophet là một mô hình dự báo dữ liệu chuỗi thời gian dựa trên mô hình cộng tính trong đó các xu hướng phi tuyến tính phù hợp với tính thời vụ hàng năm, hàng tuần và hàng ngày, cộng với các hiệu ứng ngày lễ. Nó hoạt động tốt nhất với chuỗi thời gian có tác động mạnh mẽ theo mùa và một số mùa dữ liệu lịch sử. Prophet có khả năng xử lý tốt dữ liệu bị thiếu và những thay đổi trong xu hướng, đồng thời thường xử lý tốt các ngoại lệ. [ Trích <https://facebook.github.io/prophet/>-01/05/2024]

Prophet là một mô hình có mã nguồn mở của Facebook .Nó thể hiện tốt là một mô hình về chuỗi thời gian có nhiều thời vụ và không mắc các nhược điểm của các mô hình khác

Cốt lõi của mô hình được thể hiện ở công thức sau tăng trưởng g(t), thời vụ s(t), ngày lễ h(t)và lỗi e(t)

y(t) = g(t) + s(t) + h(t) + e(t)

Trong đó:

y(t): Nhu cầu dự báo tại thời điểm t.

g(t): Mức độ cơ bản tại thời điểm t.

s(t): Thành phần mùa vụ tại thời điểm t.

h(t): Ảnh hưởng của sự kiện tại thời điểm t.

e(t):Lỗi dự báo.

***2.1.2 Hàm tăng trưởng***

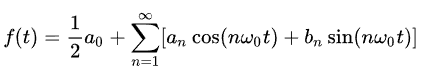
**[1]** Hàm tăng trưởng mô hình hoá xu hướng tổng thể của dữ liệu,ý tưởng được đưa vào Prophet là xu hướng tăng trưởng có thể xuất hiện ở tất cả các điểm trong dữ liệu hoặc có thể được tự đặt Prophet gọi là "thay đổi điểm".

Hàm tăng trưởng có ba tuỳ chọn chính:

* Tăng trưởng tuyến tính: một tập hợp các phương trình tuyến tính từng đoạn với độ dốc khác nhau giữa các điểm thay đổi. Thuật ngữ "tăng trưởng" tương đương với hệ số dốc trong phương trình tuyến tính cổ điển y = mx + b, tuy nhiên, ở đây, hệ số dốc và hệ số điều chỉnh sẽ biến đổi tại mỗi điểm chuyển đổi.
* Tăng trưởng logistic:tương tự như việc mô tả một phương trình điển hình cho đường cong logistic, nơi mà khả năng mang (C) biến đổi theo thời gian và tốc độ tăng trưởng (k) cùng với phần bù (m) sẽ thay đổi, tạo ra một diễn biến đa dạng và phong phú tại mỗi điểm chuyển đổi.
* Phẳng: chọn một xu hướng ổn định trong trường hợp không có sự tăng trưởng theo thời gian, mặc dù vẫn có thể tồn tại tính thời vụ.Hàm tăng trưởng sẽ duy trì giá trị không đổi tạo ra một mô hinh tĩnh và ổn định

### 2.1.3 Hàm thời vụ

**[2]** Hàm thời vụ chỉ đơn giản là một chuỗi fourier



Hình 2.1 Công thức Chuỗi Fourier

#### Theo cách nghĩ đơn giản,chuỗi fourier, là tổng của nhiều sin và co-sin liên tiếp. Mỗi số hạng sin và co-sin được nhân với một số hệ số. Kết quả có thể gần đúng với gần như bất kỳ đường cong nào hoặc trong trường hợp của Prophet, tính thời vụ (mô hình theo chu kỳ)

### 2.1.4 Ngày lễ,sự kiện

**[3]** Các dự báo của mô hình có thể bị ảnh hưởng bởi các sự kiện lớn nên mô hình Prophet có thể điều chỉnh dự báo khi xuất hiện ngày lễ hoặc sự kiện lớn có thể thay đổi dự báo.Bạn có thể tự định nghĩa các ngày lễ phù hợp với mình.

**[1][2][3]** – [Trích [Phân tích chuỗi thời gian với prophet](https://ichi.pro/vi/phan-tich-chuoi-thoi-gian-voi-facebook-prophet-cach-thuc-hoat-dong-va-cach-su-dung-no-265389504990778) – 01/05/5024 ]

### 2.1.5 Ưu điểm,nhược điểm

- Ưu điểm:

1. Dễ sử dụng: được thiết kế để dễ triển khai
2. Xử lý dữ liệu cấp cao: Prophet có khả năng xử lý dữ liệu chuỗi thời gian mà không cần nhiều tiền xử lý phức tạp có thể xử lý dữ liệu với các giá trị bị thiếu, ngoại lệ và các vấn đề phổ biến khác mà không cần nhiều can thiệp
3. Tích hợp thông tin chu kỳ: tự động tính toán và tích hợp các yếu tố chu kỳ như ngày trong tuần, tháng, năm lịch và các ngày lễ quan trọng.
4. Điều chỉnh linh hoạt: có thể tinh chỉnh các siêu tham số của mô hình để điều chỉnh hiệu suất dự đoán theo nhu cầu
5. Hỗ trợ cho dữ liệu hàng loạt: hỗ trợ dự đoán cho các chuỗi thời gian hàng loạt, làm cho nó phù hợp cho việc dự đoán trên nhiều loại dữ liệu cùng một lúc

- Nhược điểm:

1. Giới hạn về linh hoạt: Mặc dù rất mạnh mẽ cho các dự đoán chuỗi thời gian đơn giản, nhưng không đủ linh hoạt cho các trường hợp phức tạp và các yêu cầu tuỳ chỉnh cao hơn.
2. Không thích hợp cho tất cả các loại dữ liệu:không hiệu quả cho các loại dữ liệu không tuân thủ theo các mẫu chu kỳ hoặc không có xu hướng dễ nhận biết.
3. Hiệu suất dự đoán: kết quả dự đoán tốt cho các chuỗi thời gian đơn giản, nhưng hiệu suất của nó có thể giảm khi đối mặt với dữ liệu phức tạp có nhiều yếu tố gây nhiễu.
4. Không phù hợp cho các chuỗi thời gian ngắn: không phù hợp cho các chuỗi thời gian có độ dài ngắn hoặc thiếu dữ liệu lịch sử.
5. Không có khả năng dự đoán đa biến: chỉ có thể dự đoán cho một biến duy nhất, không hỗ trợ dự đoán đa biến.

## 2.2 Mô hình Random Forest

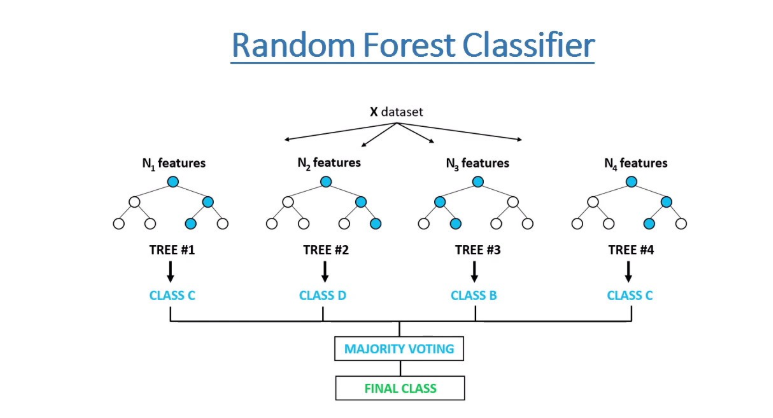
### 2.2.1 Khái niệm

**[1]** là một thuật toán học máy có giám sát phổ biến, trong đó có biến mục tiêu được gắn nhãn.

Random Forest có thể được sử dụng để giải quyết các vấn đề hồi quy và phân loại,là một phương pháp tổng hợp các dự đoán của các mô hình khác

Một mô hình trong quần thể rừng ngẫu nhiên là một cây quyết định.

Random Forest là một mô hình ước tính meta phù hợp với một số phân loại cây quyết định trên các mẫu phụ khác nhau của tập dữ liệu và sử dụng tính trung bình để cải thiện độ chính xác dự đoán và kiểm soát độ khớp quá mức.



Hình 2.2 Mô hình Random Forest Classifier

### 2.2.2 Ý tưởng của Random Forest

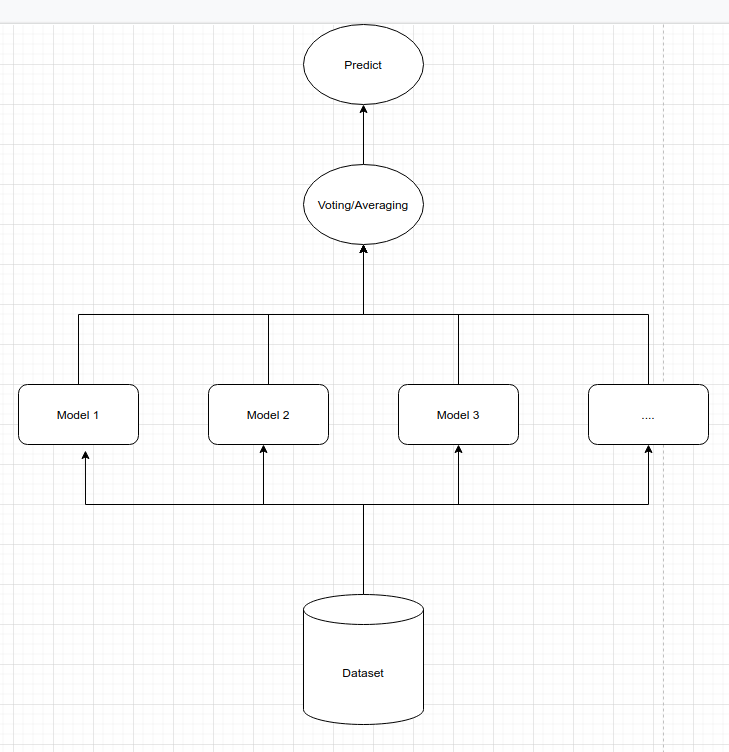
**[2]** Mô hình được huấn luyện dựa trên sự phối hợp giữa luật kết hợp và quá trình lấy mẫu tái lặp

Thuật toán của nó là tạo ra nhiều cây quyết định mà mỗi cây quyết định được huấn luyện dựa trên các tập con khác nhau kết quả dự báo là bầu cử (voting) hoặc tính toán trung bình từ toàn bộ những cây quyết định.

Kết quả được lấy từ nhiều mô hình nên sẽ không bị chệch và phương sai nhỏ hơn. Khắc phục được hiện tượng quá khớp.

### 2.2.3 Kết hợp nhiều mô hình(ensemble model)

**[3]** Ví dụ: mô hình phân loại con vật dựa trên ảnh cừu và sói lần có hai nhãn 0 và 1. Một ảnh được dự đoán trên 1 mô hình có kết quả là 0.6 là cừu.Đây là kết quả không thể chắc chắn được.Bởi vì không chắc chắn thì nên ta sẽ đi xây dựng thêm 7 mô hình nữa để voting.Ví dụ như do đây là một trường hợp khó như ảnh bị nhoè nên các mô hình như đoán kết quả không quá gần 1. dự báo từ 7 mô hình thì có 5 mô hình có kết quả là 0 và 2 mô hình còn lại là 1. Suy ra căn cứ vào kết quả voting thì ta sẽ lựa bức ảnh cừu là đúng



Hình 2.3 Mô hình kết hợp

Có nghĩa là kết hợp nhiều mô hình lại để dự đoán và sẽ chọn kết quả được voting cao nhất

Ví dụ: Xây dựng mô hình kết hợp bằng python

“from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import VotingClassifier

from sklearn.model\_selection import RepeatedStratifiedKFold

from sklearn.model\_selection import cross\_val\_score

from sklearn.metrics import accuracy\_score

from sklearn.datasets import load\_iris

from sklearn.datasets import make\_blobs

import numpy as np

iris = load\_iris()

X = iris.data

y = iris.target = = 1

learninglog\_clf = LogisticRegression()

svm\_clf = SVC()

tree\_clf = DecisionTreeClassifier(max\_depth=3)

voting\_clf = VotingClassifier(

estimators=[('lr', log\_clf), ('svc', svm\_clf), ('tree\_clf', tree\_clf)],

voting='hard')

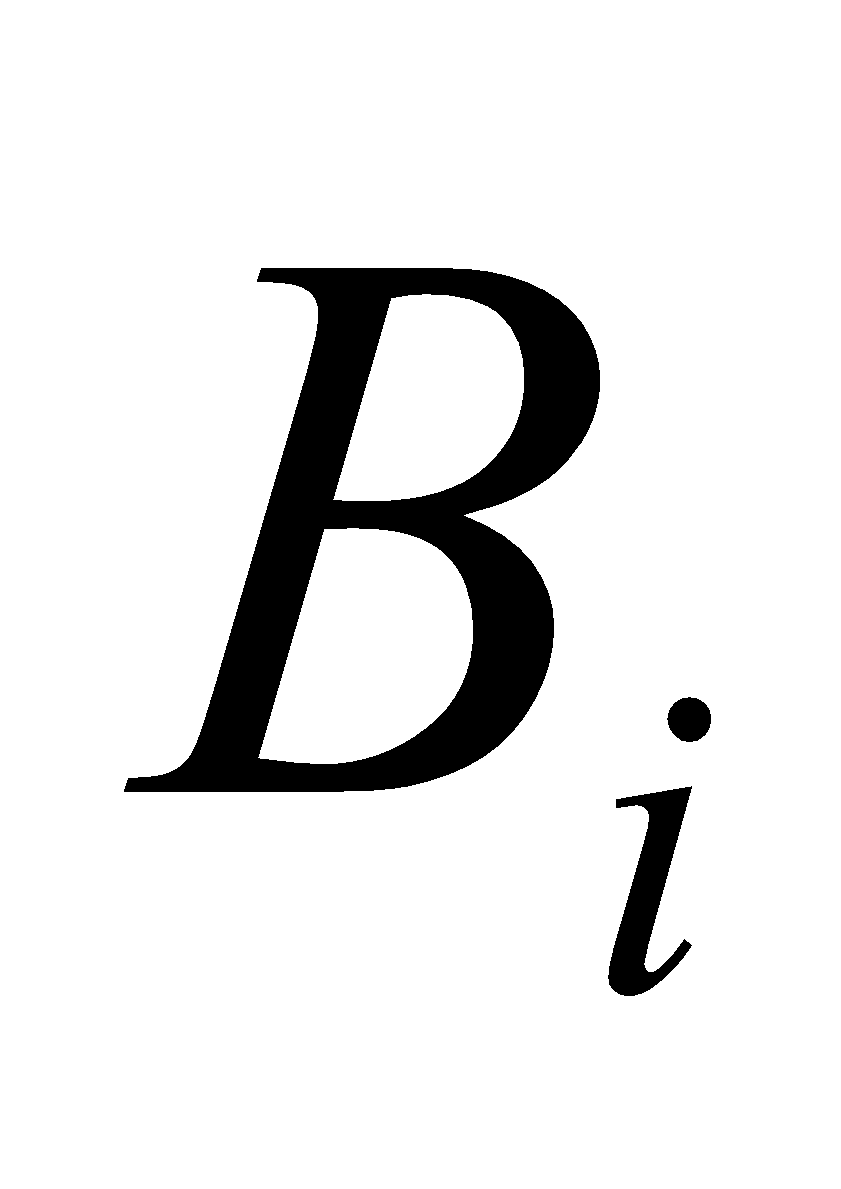
cv = RepeatedStratifiedKFold(n\_splits=3, n\_repeats=5, random\_state=1)”

### 2.2.4 Lấy mẫu (boostrapping)

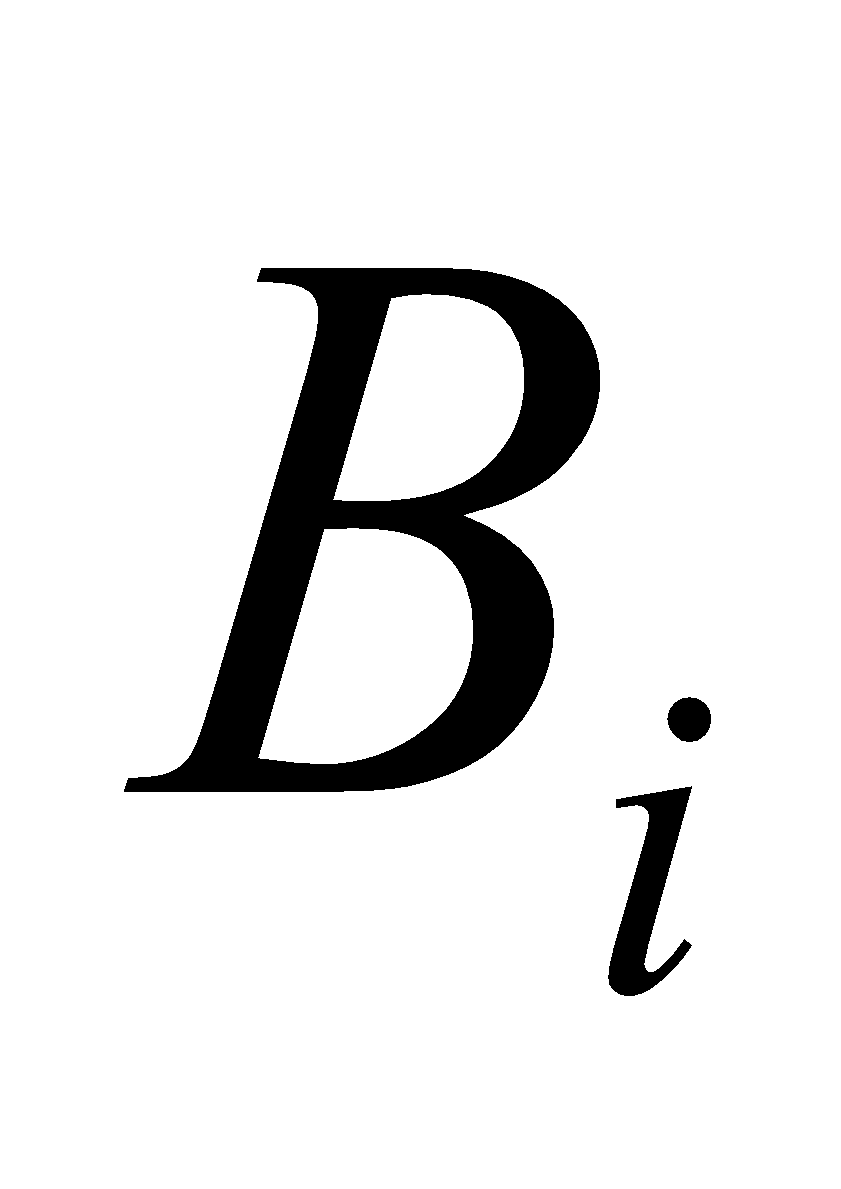
**[4]** Ví dụ: cho tập hợp là tập dữ liệu huấn luyện

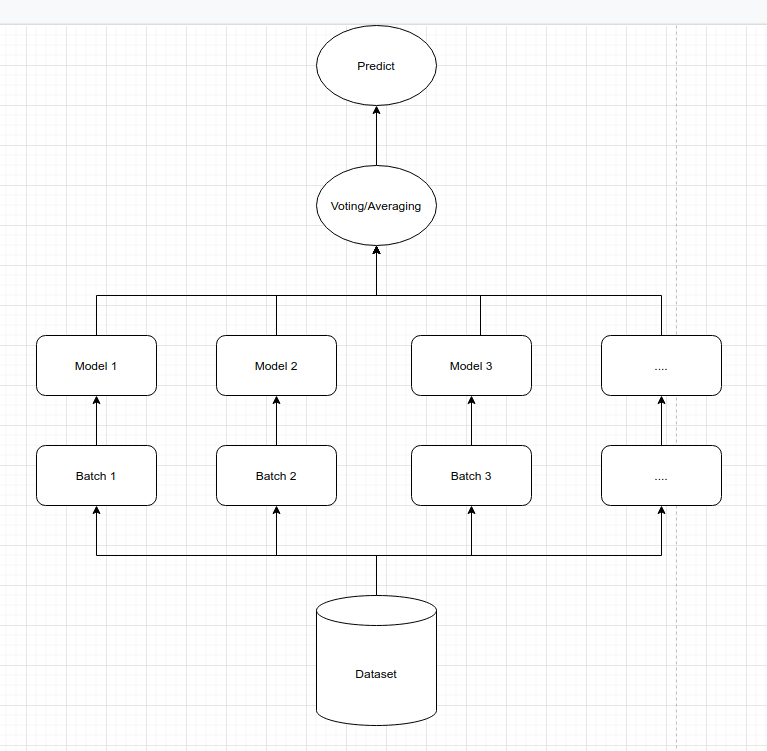


bao gồm các quan sát.Thuật toán sẽ sử dụng phương pháp lấy mẫu tái lập để tạo thành B tập dữ liệu con. Quá trình này còn gọi là bỏ túi (bagging)

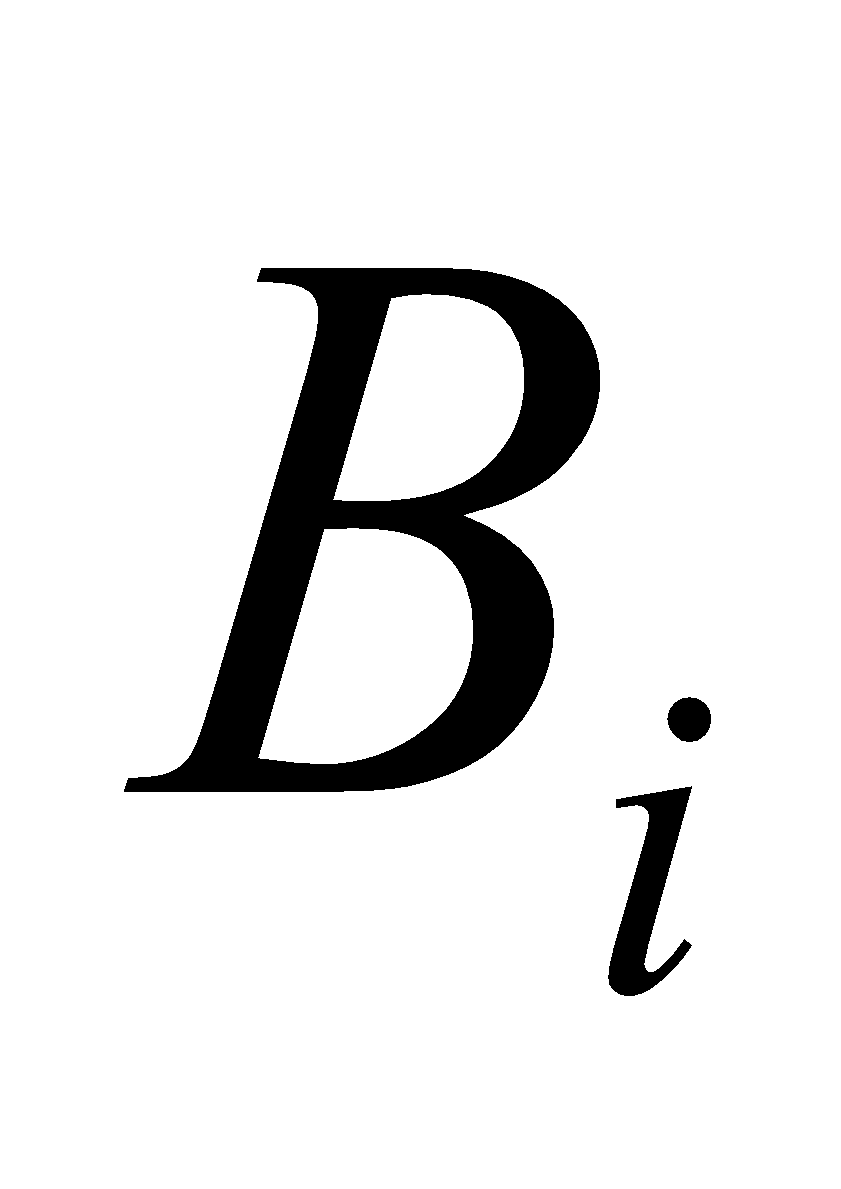
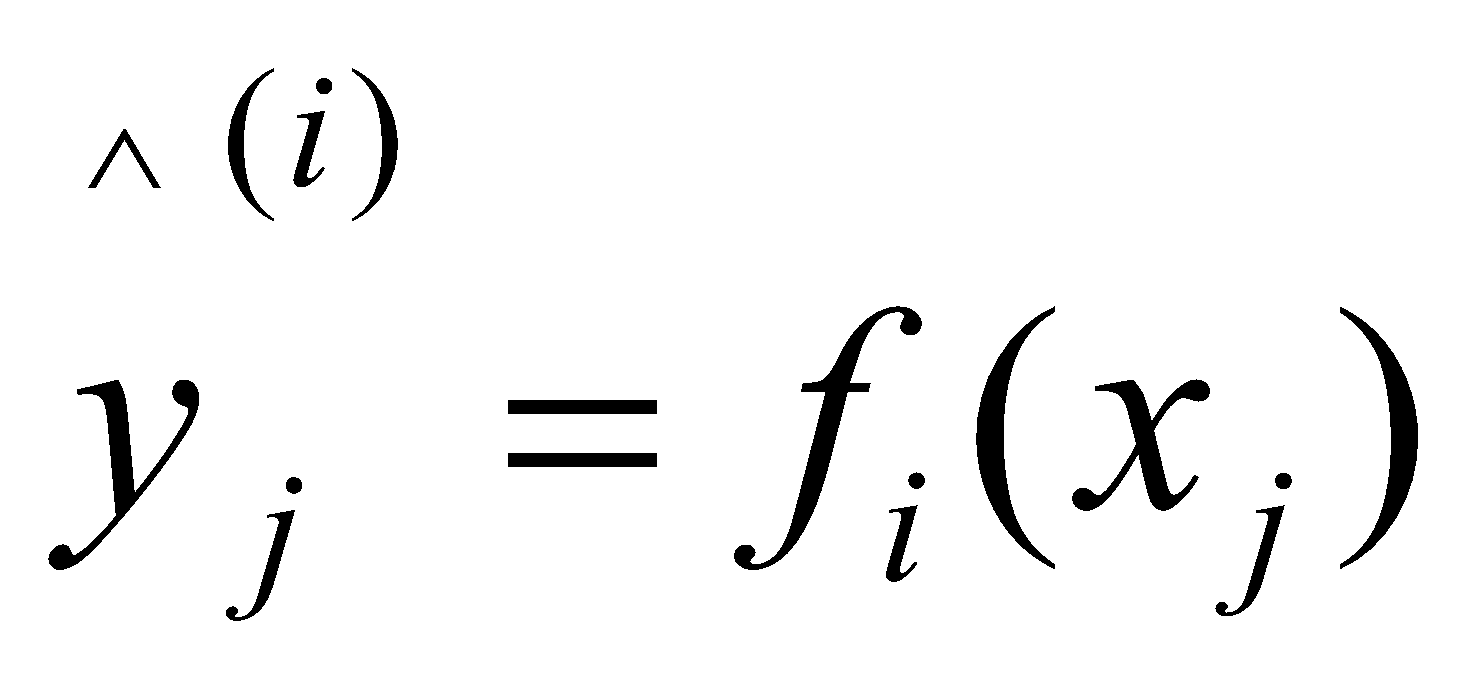
Tức là chúng ta sẽ thực hiện M lượt nhặt các mẫu từ tổng thể và bỏ vào túi để tạo thành tập.Tập  cho phép các phần tử được lặp lại



Như vậy sẽ tồn tại những quan sát D không thuộc  đây là những quan sát chưa được bỏ vào túi gọi chúng là nằm ngoài túi(out of bag)



#### Hình 2.4 Lấy mẫu tái lập

Với mỗi tập dữ liệu  ta xây dựng một mô hình cây quyết định trả về kết quả dự báo là 

Trong đó :

là kết quả dự báo



là giá trị đầu vào



là hàm dự báo của mô hình



-Mô hình dự báo:tính giá trị trung bình của các dự báo từ mô hình con.



-Mô hình phân loại: thực hiện bầu cử từ các mô hình con để chọn ra nhãn dự báo có tần suất lớn nhất.



Phương sai trong mô hình dự đoán



Bởi vì dự báo của các mô hình con không bị ảnh hưởng hoặc liên quan đến nhau nên có thể suy ra kết quả dự báo là độc lập . Tức là



Đồng thời giả định chất lượng các mô hình là đồng đều, được thể hiện qua phương sai dự báo là đồng nhất .Từ đó suy ra:



Dự báo là trung bình kết hợp từ nhiều mô hình cây quyết định thì phương sai có thể giảm nhiều so với chỉ sử dụng một mô hình

Trong mô hình rừng cây, số lượng các cây quyết định là rất lớn. Nên phương sai dự báo từ mô hình có thể giảm gấp nhiều lần và tạo ra một dự báo tốt hơn

Ví dụ: Sử dụng module BaggingClassifier để áp dụng thuật toán bỏ túi

“from sklearn.ensemble import BaggingClassifier

from sklearn.tree import DecisionTreeClassifier

bag\_clf = BaggingClassifier(

DecisionTreeClassifier(),

n\_estimators=200,

max\_samples=100,

bootstrap=True,

n\_jobs=-1)

bag\_clf.fit(X, y)”

### 2.2.5 Đánh giá mô hình bằng mẫu nằm ngoài túi(out of bag)

**[5]** Phương pháp bỏ túi cho phép lấy một số lượng lớn tập con huấn luyện cho phép lặp lại nên con một số lượng lớn các mẫu không lấy. Tập hợp những mẫu này gọi là mẫu nằm ngoài túi (out of bag), được viết tắt là oob.

Những tập dữ liệu này được lựa chọn ngẫu nhiên và độc lập và không được học trong mô hinh nên có thể sử dụng để kiểm tra mô hình

Trong python, BaggingClassifier có thể sử dụng thêm tuỳ chọn oob\_score=True để đánh giá mô hình dựa trên các mẫu oob.

Ví dụ:gọi oob\_score để đánh giá mô hình dựa trên các mẫu oob

“from sklearn.ensemble import BaggingClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

idx = np.arange(X.shape[0])

np.random.seed(0)

np.random.shuffle(idx)

idx\_train = idx[:100]

idx\_test = idx[100:]

X\_train, y\_train = X[idx\_train, :], y[idx\_train]

X\_test, y\_test = X[idx\_test, :], y[idx\_test]

bag\_clf = BaggingClassifier(

DecisionTreeClassifier(),

n\_estimators=200,

max\_samples=100,

bootstrap=True,

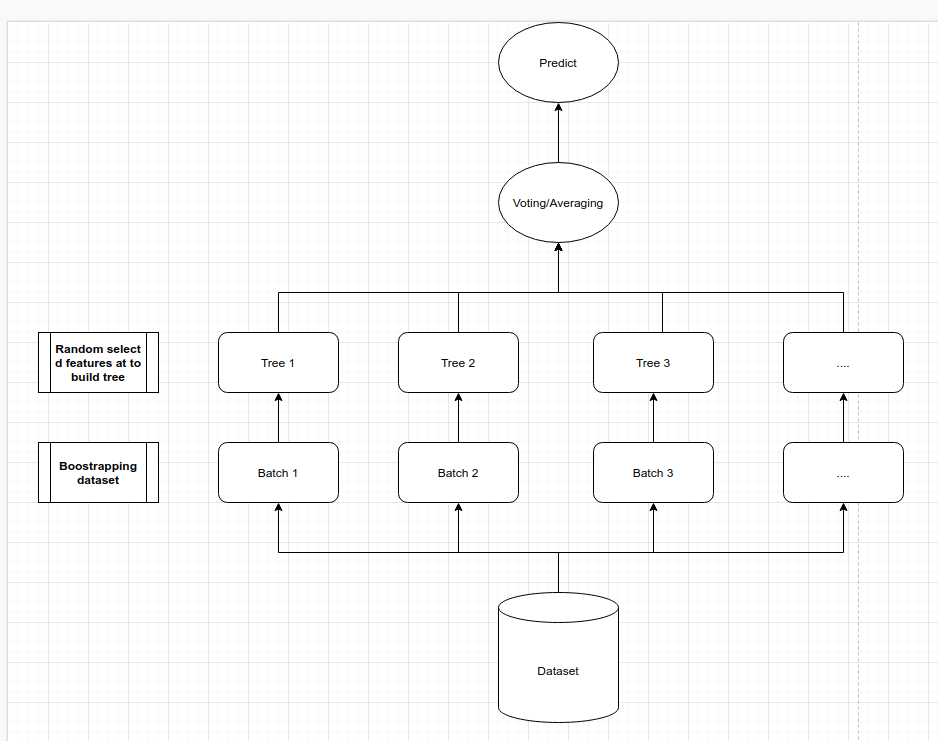
oob\_score=True,

n\_jobs=-1)

bag\_clf.fit(X\_train, y\_train)”

### 2.2.6 Mô hình Random Forest

**[6]** Mô hình rừng cây sẽ áp dụng cả hai phương pháp học kết hợp (ensemble learning) và lấy mẫu tái lập (boostrapping)



#### Hình 2.5 Sơ đồ mô hình cây quyết định

Thứ tự của quá trình tạo thành một mô hình rừng cây như sau:

1. Từ tập huấn luyện lấy mẫu tái lập ngẫu nhiên tạo một tập dữ liệu con
2. Chọn lọc các đặc trưng và bắt đầu xây dựng mô hình,ở đây bước 1 và bước 2 được thực hiện nhiều lần vì ta sẽ đi xây dựng nhiều mô hình gọi là cây quyết định
3. Đưa ra dự báo băng cách bầu cử hoặc tính trung bình

Mô hình rừng cây được huấn luyện trên nhiều tập dữ liệu con khác nhau,có một số tập dữ liệu có ngoại lai điều này giúp cho mô hình ít bị nhạy cảm với dữ liệu ngoại lai hơn.

Bộ dữ liệu được sử dụng để học là một nên quy luật học được giữa các cây quyết định sẽ gần tương tự như nhau và tổng hợp kết quả giữa chúng không có xu hướng bị chệch.

### 2.2.7 Huấn luyện mô hình RandomForest

**[7]**Trong bài toán phân loại ta sử dụng class [RandomForestClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) và bài toán dự báo là class [RandomForestRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html).(python)



#### Hình 2.6 Các tham số của mô hình

Bản chất của mô hình rừng cây là sự kết hợp giữa nhiều cây quyết định được huấn luyện theo phương pháp lấy mẫu tái lập.

Trong đó :

n\_estimators là số lượng các cây quyết định được sử dụng trong mô hình rừng cây.

bootstrap=True tương ứng với sử dụng phương pháp lấy mẫu tái lập khi xây dựng các cây quyết định. Trái lại thì chúng ta sử dụng toàn bộ dữ liệu.

oob\_score chỉ có hiệu lực khi sử dụng khi sử dụng boostrap. Nếu oob\_score=True thì sẽ tính toán thêm điểm số trên các mẫu nằm ngoài túi.

max\_samples là số lượng mẫu được sử dụng để huấn luyện mô hình cây quyết định. Mặc định max\_samples=None thì chúng ta lấy ra các mẫu con có kích thước bằng với tập huấn luyện.

“Ví dụ:Huấn luyện mô hình rừng cây.

from sklearn.pipeline import Pipeline

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

rdf\_clf = RandomForestClassifier(

max\_depth = 3,

max\_leaf\_nodes = 16,

min\_samples\_split = 10,

min\_samples\_leaf = 10)

rdf\_clf.fit(X\_train, y\_train)”

[1][2][3][4][5][6][7] – [Trích <https://phamdinhkhanh.github.io/deepai-book/ch_ml/RandomForest.html> -01/05/2024]

# ONE-HOT-ENCODING,LABEL ENCODER

## 3.1 One-Hot-Encoding

### 3.1.1 Khái niệm

**[1]** One Hot Encoding là một quy trình xử lý dữ liệu được áp dụng cho dữ liệu phân loại, để chuyển đổi nó thành một biểu diễn véc-tơ nhị phân để sử dụng trong các thuật toán học máy

 One Hot Encoding , biểu diễn mảng véc-tơ nhị phân cho phép thuật toán máy học tận dụng thông tin chứa trong giá trị danh mục mà không gây ra sự nhầm lẫn do thứ tự .

***3.1.2 Quy trình hoạt động***

1. Mỗi giá trị của biến phân loại được chuyển thành một biến giả mới.
2. Mỗi mẫu dữ liệu trong tập dữ liệu sẽ được biểu diễn bằng một véc-tơ có độ dài bằng số lượng giá trị duy nhất của biến phân loại.
3. Trong vector chỉ có một phần tử sẽ có giá trị là 1 (đại diện cho giá trị của biến phân loại của mẫu dữ liệu đó), và các phần tử còn lại sẽ có giá trị là 0.



#### Hình 3.1 Ví dụ minh hoạ mã hoá one-hot

[1]- [Trích <https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html>-01/05/2024 ]

### 3.1.3 Ưu điểm,nhược điểm

-Ưu điểm:

1. [Cho phép biểu diễn dữ liệu hạng mục dưới dạng số](https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html)
2. [Mỗi giá trị hạng mục được biểu diễn bằng một vector nhị phân, giúp cho việc phân biệt giữa các hạng mục trở nên dễ dàng](https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html)
3. Hiệu quả về mặt tính toán: Việc mã hoá không tốn nhiều thời gian và tài nguyên tính toán.
4. Giữ nguyên thứ tự: Thứ tự của các giá trị trong biến danh mục được bảo toàn sau khi mã hoá.

-Nhược điểm:

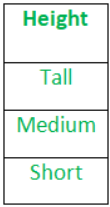
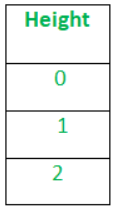
1. [Kích thước của dữ liệu có thể tăng lên đáng kể khi sử dụng mã hóa one-hot, đặc biệt là khi có nhiều giá trị hạng mục](https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html).
2. [Mã hóa one-hot không thể xử lý các giá trị hạng mục mới không có trong từ điển đã xây dựng](https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html)
3. Mất thông tin về mối quan hệ: không thể hiện mối quan hệ giữa các giá trị trong biến danh mục
4. Có thể tạo ra dữ liệu thưa thớt: khi có nhiều giá trị duy nhất trong biến danh mục, dữ liệu sau khi mã hoá có thể trở nên rất thưa thớt

## 3.2 Label Encoder

### 3.2.1 Khái niệm

Label Encoder là một lớp trong thư viện scikit-learn được sử dụng để chuyển đổi dữ liệu phân loại thành dạng số.Để có thể tính toán được trên mô hình máy học

Chưa chuyển đổi =========> Sau khi chuyển đổi

#### Hình 3.2 Ví dụ minh hoạ mã hoá Label

### 3.2.2 Quy trình hoạt động

1. Tạo một đối tượng Label Encoder
2. Gọi phương thức fit\_transfrom để mã hoá dữ liệu

Ví dụ: Mã hoá **Label Encoder** bằng python

“from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

data = ["màu đỏ", "xanh lam", "tốt", "màu đỏ", "xanh lam"]

encoded\_data = encoder.fit\_transform(data)”

### 3.2.3 Ưu điểm,nhược điểm

-Ưu điểm:

1. Dễ dàng sử dụng và hiểu
2. Hiểu quả về mặt tính toán
3. Sử dụng được với cả dữ liệu loại nhị phân và đa phân loại

-Nhược điểm:

1. Không đảm bảo được thứ tự tự nhiên của các nhãn
2. Có thể ảnh hưởng đến hiệu suất của một số mô hình máy học
3. Chỉ định một số duy nhất (bắt đầu từ 0) cho mỗi lớp dữ liệu.

# TINH CHỈNH THAM SỐ MÔ HÌNH VÀ LỰA CHỌN ĐẶC TRƯNG

## 4.1 Parameter Grid

### 4.1.1 Khái niệm

### **Parameter Grid** (lưới tham số) là tập hợp các giá trị được sử dụng để điều chỉnh hiệu suất của một mô hình máy học bao gồm các giá trị khác nhau cho các siêu tham số, là những tham số cấu hình ảnh hưởng đến cách thức hoạt động của mô hình.

Các kết hợp giá trị trong lưới tham số sẽ được mô hình học máy thử nghiệm tất cả để tìm ra bộ tham số tốt nhất cho mô hình

Ví dụ: Cách tạo ra lưới siêu tham số cho mô hình học máy

“from sklearn.model\_selection import ParameterGrid

param\_grid = {'a': [1, 2], 'b': [True, False]}

list(ParameterGrid(param\_grid)) = = (

[{'a': 1, 'b': True}, {'a': 1, 'b': False},

{'a': 2, 'b': True}, {'a': 2, 'b': False}])”

### 4.1.2 Mục đích của Parameter Grid

Tìm giá trị tối ưu cho siêu tham số: giúp xác định giá trị nào mang lại hiệu quả tốt nhất cho mô hình trên một tập dữ liệu cụ thể.

Đánh giá hiệu suất của mô hình: đánh giá hiệu suất của mô hình dưới các cấu hình khác nhau, cung cấp thông tin chi tiết về cách mô hình phản ứng với những thay đổi trong các siêu tham số.

So sánh các mô hình: có thể được dùng để so sánh hiệu suất của các mô hình khác nhau được đào tạo với cùng một tập dữ liệu nhưng với các siêu tham số khác nhau.

### 4.1.3 Cách sử dụng Parameter Grid

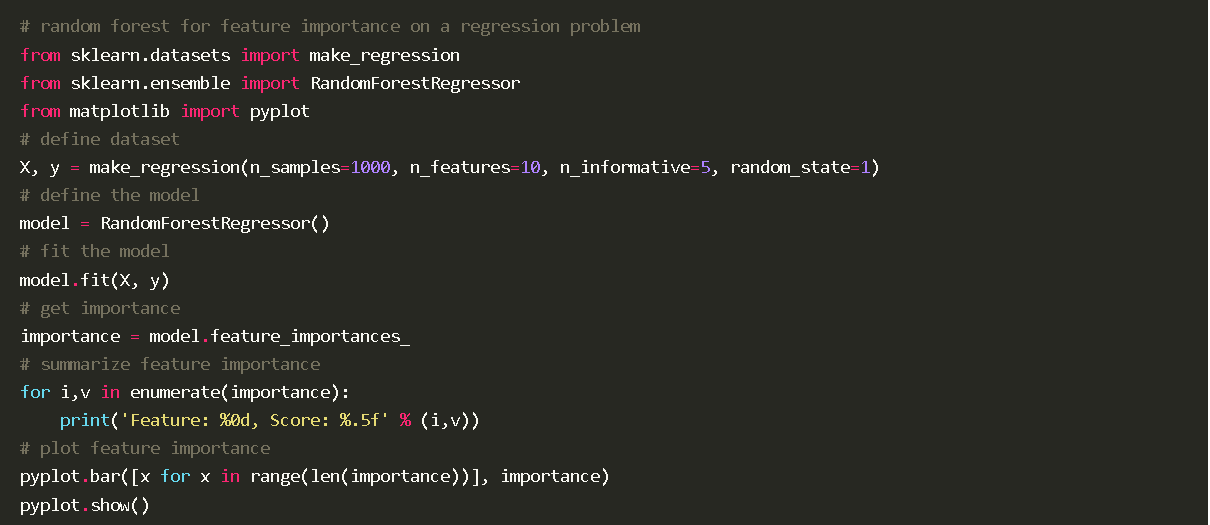
Parameter Grid thường được sử dụng kết hợp với các kỹ thuật tối ưu hoá, chẳng hạn như tìm kiếm ngẫu nhiên hoặc tìm kiếm theo hướng dẫn, để tự động tìm giá trị tối ưu cho các siêu tham số. Quá trình này có thể bao gồm nhiều bước lặp, trong đó mô hình được đào tạo với một tập hợp các siêu tham số, hiệu suất được đánh giá và sau đó các siêu tham số được điều chỉnh dựa trên kết quả.

## 4.2 Thuộc tính feature\_importance\_

### 4.2.1 Khái niệm

Feature\_importance\_ là một thuộc tính của mô hình được sử dụng để hiển thị mức độ quan trọng của các đặc trưng ,giá trị này thể hiện mức độ ảnh hưởng của mỗi đặc trưng đến biến mục tiêu (target variable) mà mô hình đang cố gắng dự đoán.

Thuộc tính feature\_importance\_ thường được hỗ trợ bởi một số mô hình học máy có đặc điểm nhất định như một số mô hình như Decision Trees,Random Forests,Gradient Boosting Machines,Extra Trees,Ada Boost



#### Hình 4.1 Ví dụ sử dụng thuộc tính feature\_importance\_

### 4.2.2 Cách tính feature\_importance\_

Có các phương pháp khác nhau để tính toán feature\_importance\_, phương pháp cụ thể sẽ phụ thuộc vào thuật toán học máy học mà ta sử dụng.

Một số phương pháp phổ biến:

* **Mean Decrease Impurity:** tính toán mức độ giảm thiểu độ "không tinh khiết" trung bình của một nút cây quyết định khi một đặc trưng bị xáo trộn.
* **Permutation Importance:** xáo trộn giá trị của một đặc trưng và đo lường sự thay đổi hiệu suất của mô hình.
* **Coefficients:** một số mô hình tuyến tính, feature\_importance\_ có thể được tính toán trực tiếp từ hệ số hồi quy.

### 4.2.3 Mục đích của feature\_importance\_

1. Lựa chọn đặc trưng: Xác định những đặc trưng quan trọng nhất và loại bỏ những đặc trưng không quan trọng, giúp giảm thiểu kích thước mô hình và cải thiện hiệu suất.
2. Hiểu mô hình: Hiểu rõ hơn về cách mô hình đưa ra dự đoán và xác định những đặc trưng nào ảnh hưởng nhất đến kết quả.
3. Truyền thông kết quả: Giải thích kết quả của mô hình cho người khác bằng cách nêu bật những đặc trưng quan trọng nhất.

# ĐÁNH GIÁ HIỆU SUẤT MÔ HÌNH

## 5.1 .Cross-validation trong prophet

Prophet bao gồm chức năng xác thực chéo chuỗi thời gian để đo lường lỗi dự báo bằng cách sử dụng dữ liệu lịch sử. Điều này được thực hiện bằng cách chọn các điểm cắt trong lịch sử và đối với mỗi điểm cắt phù hợp với mô hình chỉ sử dụng dữ liệu cho đến điểm cắt đó. Sau đó chúng ta có thể so sánh các giá trị dự báo với giá trị thực tế.

Quy trình xác thực chéo được thực hiện tự động đối với một loạt các điểm cắt lịch sử bằng cách sử dụng cross\_validation trong đó có thể chỉ định khoảng thời gian dự báo ( horizon),chọn quy mô của giai đoạn đào tạo ban đầu ( initial) và khoảng cách giữa các ngày giới hạn ( period).

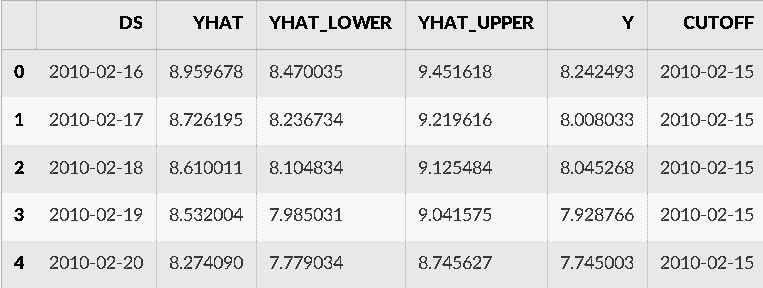
Đầu ra của cross\_validation là một khung dữ liệu có các giá trị thực y và các giá trị dự báo ngoài mẫu yhat tại mỗi ngày dự báo được mô phỏng và cho mỗi ngày giới hạn. Đặc biệt, dự báo được thực hiện cho mọi điểm được quan sát giữa cutoff và cutoff + horizon. Và khung dữ liệu này có thể được sử dụng để tính toán các thước đo lỗi yhat so với y.

Ví dụ:Sử dụng cross\_validation

“from prophet.diagnostics import cross\_validation

df\_cv = cross\_validation(m, initial='730 days', period='180 days', horizon = '365 days') “

m được hiểu là mô hình prophet

Hình 5.1 Đầu ra của cross\_validation

Các điểm cắt tuỳ chỉnh cũng có thể được cung cấp dưới dạng danh sách ngày bằng từ khoá cutoffs trong cross\_validation

Ví dụ:Cắt 3 ngày trong data và khai báo bằng cutoffs trong Python

“cutoffs = pd.to\_datetime(['2013-02-15', '2013-08-15', '2014-02-15'])

df\_cv = cross\_validation(m, cutoffs=cutoffs, horizon='365 days')”

m được hiểu là mô hình prophet

## 5.2 Song song trong cross\_validation

Xác thực chéo cũng có thể được chạy ở chế độ song song trong Python, bằng cách đặt chỉ định từ khoá parallel. Bốn chế độ được hỗ trợ:

1. parallel=None(Mặc định, không song song)
2. parallel="processes"
3. parallel="threads"
4. parallel="dask"

Đối với những vấn đề không quá lớn, nên sử dụng parallel="processes". Nó sẽ đạt được hiệu suất cao nhất khi việc xác thực chéo song song có thể được thực hiện trên một máy duy nhất. Đối với các vấn đề lớn, cụm [Dask](https://dask.org/) có thể được sử dụng để thực hiện xác thực chéo trên nhiều máy. Nhưng phải [cài đặt Dask](https://docs.dask.org/en/latest/install.html) riêng vì nó sẽ không được cài đặt cùng với prophet.

Ví dụ:Sử dụng song song “dask” trong python

“from dask.distributed import Client

client = Client()

clusterdf\_cv = cross\_validation(m, initial='730 days', period='180 days', horizon='365 days',parallel="dask")”

## 5.3 Performance\_metrics trong prophet

Performance\_metrics có thể được sử dụng để tính toán một số thống kê hữu ích về hiệu suất dự đoán ( yhat, yhat\_lower, và yhat\_upper so sánh với y), dưới dạng hàm của khoảng cách từ điểm cắt (dự đoán đó là bao xa trong tương lai).

Các số liệu thống kê được tính toán là sai số bình phương trung bình (MSE), sai số bình phương trung bình gốc (RMSE), sai số tuyệt đối trung bình (MAE), sai số phần trăm tuyệt đối trung bình (MAPE), sai số phần trăm tuyệt đối trung bình (MDAPE) và mức độ bao phủ của yhat\_lower và yhat\_upper ước tính.

Chúng được tính toán trên một cửa sổ cuộn các dự đoán của mô hình sau khi sắp xếp theo đường chân trời ( ds - cutoff).

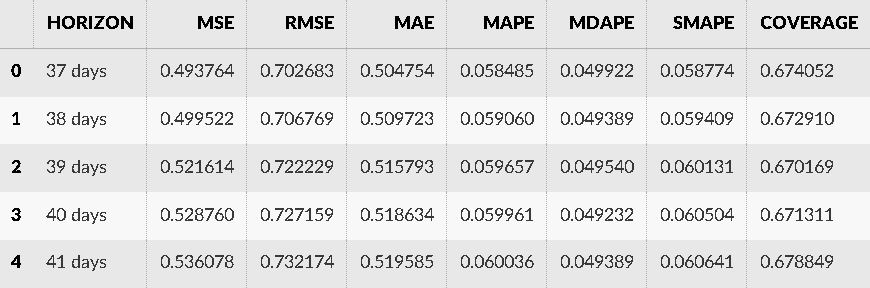
Theo mặc định, 10% dự đoán sẽ được đưa vào mỗi cửa sổ, nhưng điều này có thể được thay đổi bằng đối rolling\_window

Ví dụ: Sử dụng performance\_metrics trong python

“from prophet.diagnostics import performance\_metricsdf\_

p = performance\_metrics(df\_cv)”

df\_cv được hiểu là đầu ra của cross\_validation



#### Hình 5.2 Kết quả của performance\_metrics

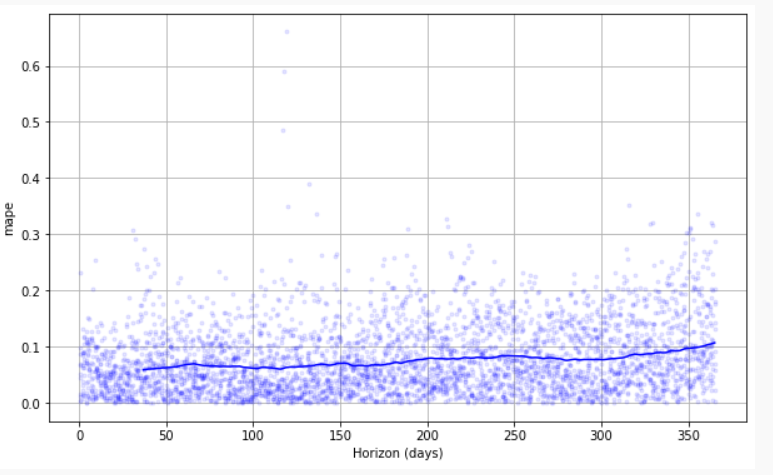
## 5.4 Plot\_cross\_validation\_metric trong prophet

Số liệu hiệu suất xác thực chéo có thể được hiển thị bằng plot\_cross\_validation\_metric. Các dấu chấm hiển thị lỗi phần trăm tuyệt đối cho mỗi dự đoán trong.Đường màu xanh lam thể hiện sai số , trong đó giá trị trung bình được lấy trên một cửa sổ cuộn gồm các dấu chấm.

Ví dụ: Sử dụng plot\_cross\_validation\_metric

“from prophet.plot import plot\_cross\_validation\_metric

fig = plot\_cross\_validation\_metric(df\_cv, metric='mape')”

Hình 5.3 Kết quả plot\_cross\_validation\_metric

# REORDER POINT,SAFETY STOCK,ECONOMIC ORDER QUANTITY - EOQ

## 6.1 Reorder point

### 6.1.1 Khái niệm

Điểm đặt hàng lại là thời điểm công ty cần bổ sung nguồn cung cấp nguyên liệu thô, linh kiện hoặc hàng hoá hoàn chỉnh.Để tránh thiếu hụt hàng hoá trong quá trình chờ đợi lô hàng mới đến

Công thức:

Điểm đặt hàng lại =Doanh số trung bình hàng ngày x Thời gian giao hàng trong ngày

Công thức khi có kết quả kho an toàn:

Điểm đặt hàng lại = Tồn kho an toàn+ (Doanh số trung bình hàng ngày x Thời gian giao hàng trong ngày)

### 6.1.2 Tầm quan trọng cảu điểm đặt hàng lại

1. Tránh tình trạng hết hàng
2. Tối ưu hoá mức tồn kho
3. Cải thiện vòng quay hàng tồn kho
4. Cải thiện dòng tiền
5. Tiết kiệm chi phí

## 6.2 Safety stock

Kho an toàn là một thuật ngữ được sử dụng trong quản lý hàng tồn kho đề cập đến mức tồn kho bổ sung được duy trì để giảm thiểu rủi ro hết hàng.

Tình trạng hết hàng có thể dẫn đến mất doanh số bán hàng, khách hàng không hài lòng và sản xuất bị đình trệ.

Công thức:

Tồn kho an toàn=[doanh số bán hàng tối đa hàng ngày x thời gian giao hàng tối đa] – [ doanh số bán hàng trung bình hàng ngày x thời gian giao hàng trung bình]

## 6.3 Economic order quantity - EOQ

Số lượng đặt hàng kinh tế (EOQ) là lượng hàng tồn kho mà doanh nghiệp nên đặt hàng để giảm thiểu chi phí tồn kho và lưu kho. Số lượng này dựa trên khối lượng bán hàng, chu kỳ sản xuất và chi phí tồn kho của công ty. EOQ là điểm mà tại đó chi phí đặt hàng và vận chuyển của công ty bằng nhau.

Chi phí đặt hàng của công ty bao gồm chi phí đặt hàng và nhận đơn đặt hàng cũng như chi phí nguyên vật liệu. Chi phí lưu kho là chi phí lưu trữ hàng tồn kho, bao gồm chi phí về mặt bằng và chi phí nguyên vật liệu.

Mô hình EOQ là một cách đơn giản để xác định số lượng đặt hàng tối ưu cho một công ty. Mô hình này tính đến khối lượng bán hàng, chu kỳ sản xuất và chi phí tồn kho của công ty. EOQ là điểm mà tại đó chi phí đặt hàng và vận chuyển của công ty bằng nhau.

Công thức:

EOQ =



Trong đó:

EOQ là kích thước đặt hàng tối ưu

D là tốc độ tiêu thụ hàng hoá (trong 1 ngày hoặc 1 tuần )

S là chi phí đặt hàng

H là chi phí lưu trữ hàng tồn kho

# XÂY DỰNG MÔ HÌNH

## Giới thiệu bài toán

### 7.1.1. Giới thiệu data

Đây là một bộ dữ liệu về chuỗi cung ứng trong 3 năm gần đây của một công ty ở Hoa Kỳ chuyên bán về quần áo,đồ thể thao và đồ điện tử.Nó có 180519 dòng và 53 cột

Trích dẫn của data: [linkdata](about:blank)

### 7.1.2. Bài toán đặt ra

**Thứ nhất:** Bài toán mà nhóm em đặt ra ở bài tiểu luận này là sản phẩm bán chạy nhất của công ty sẽ đạt được bao nhiêu doanh thu trong 12 tháng tới ?

**Thứ hai:** Chúng ta nên dự trữ tồn kho bao nhiêu là an toàn nếu có biến cố sẩy ra? Ở mức tồn kho như thế nào thì chúng ta có thể đặt hàng? Đặt bao nhiêu để tối ưu hoá chi phí?

### 7.1.3. Giải quyết vấn đề

Nhóm chúng em sẽ sử dụng mô hình học máy prophet để dự đoán doanh thu cho vấn đề thứ nhất

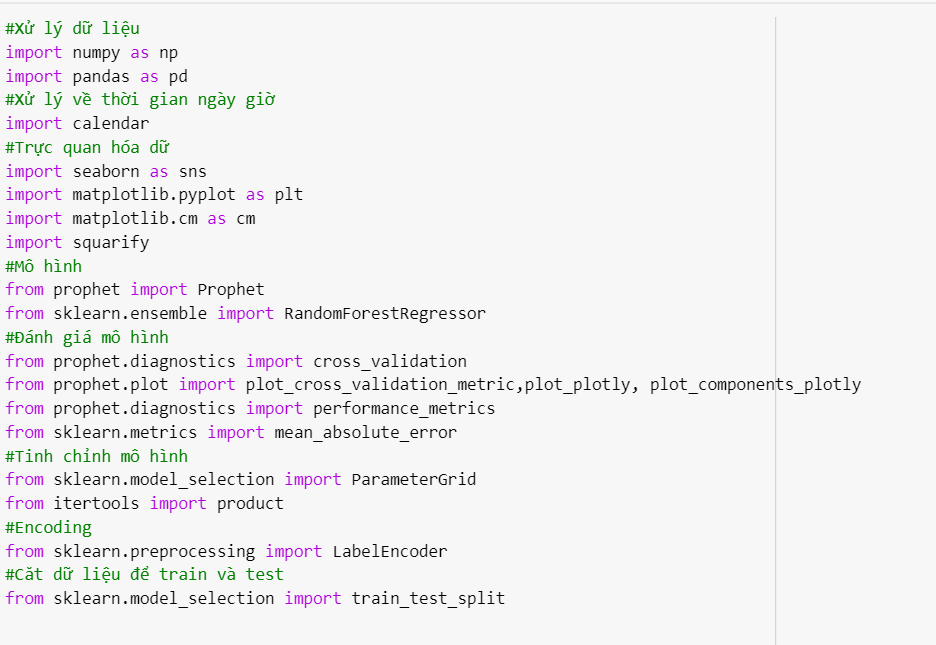
Ở vấn đề thứ hai nhóm chúng em sẽ tính số lượng điểm đặt hàng lại, lượng tồn kho an toàn và số lượng đặt hàng kinh tế (EOQ) cho sản phẩm

Chúng ta cùng đi khám phá và trả lời hai vấn đề ở trên

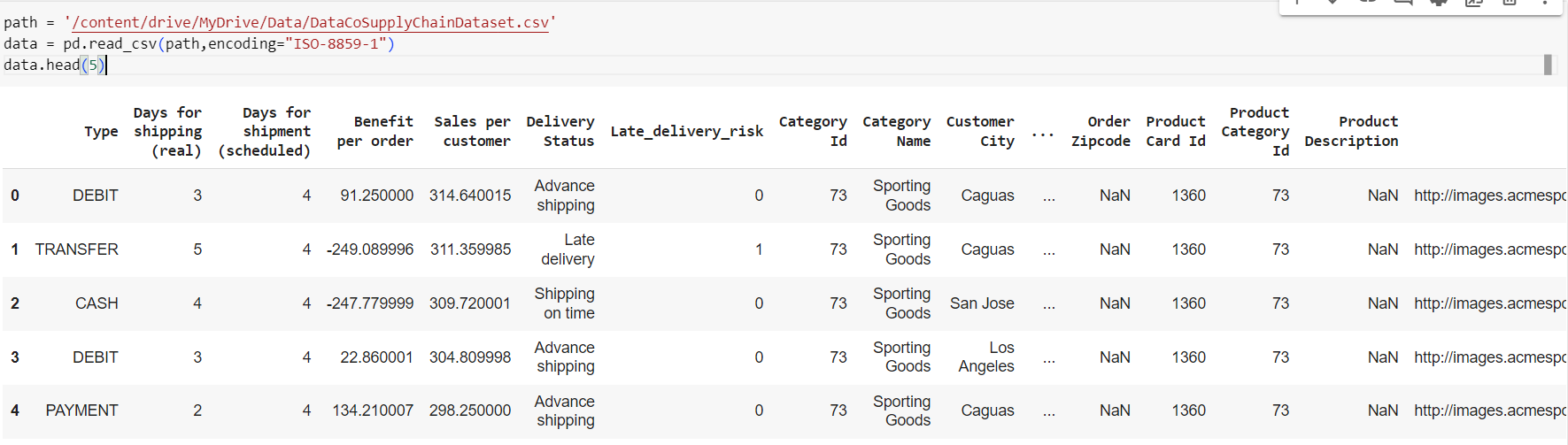
## Tìm hiểu dữ liệu và xây dựng mô hình

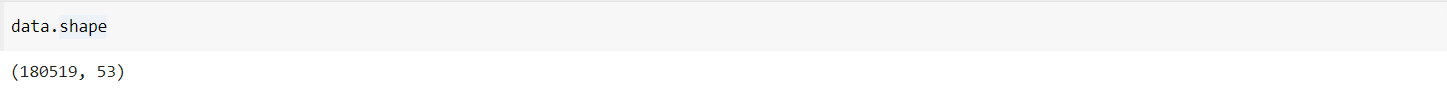
### 7.2.1 Tìm hiểu dữ liệu

-Khai báo thư viện



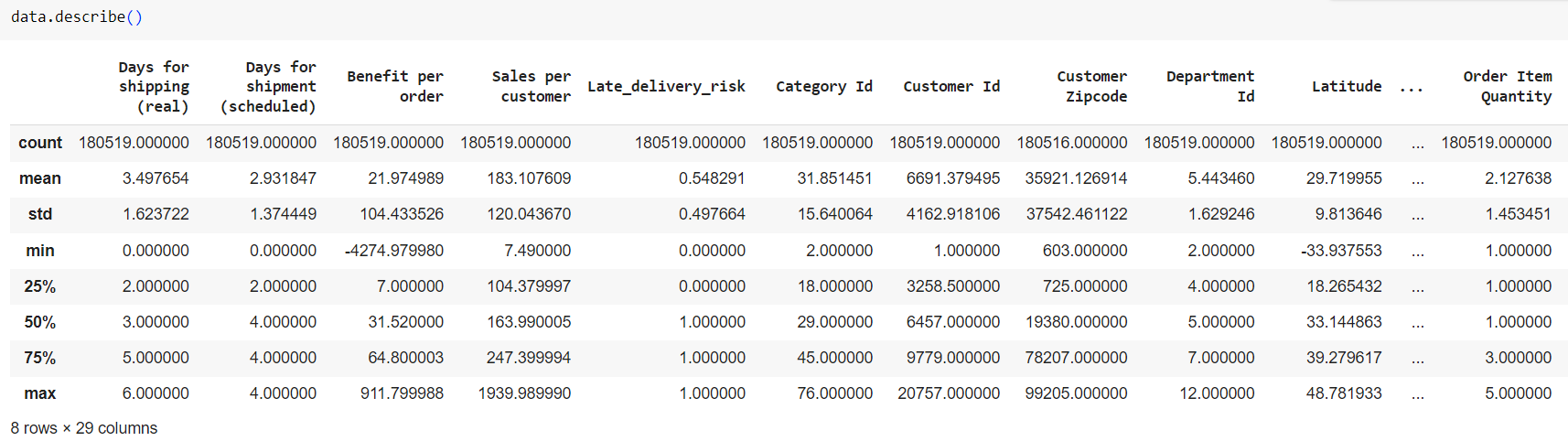
-Đọc dữ liệu





Ở đây ta thấy dữ liệu của ta có 180519 hàng và 53 cột

-Thống kê dữ liệu



### 7.2.2 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu chúng em sẽ tập trung vào dữ liệu bán hàng lịch sử và các thuộc tính sản phẩm như; mức tồn kho và danh mục sản phẩm, chúng em cũng sẽ phân tích tác động của các biến số khác góp phần tạo nên mô hình nhu cầu bao gồm các yếu tố địa lý, phân khúc khách hàng và thời gian giao hàng.

Nhiệm vụ tiền xử lý:

1. Bỏ các cột không liên quan
2. Xóa các hàng có giá trị bị thiếu,kiểm tra trùng lặp
3. Tạo đặc trưng mới
4. Chuyển đổi đặc điểm phân loại thành đặc điểm số

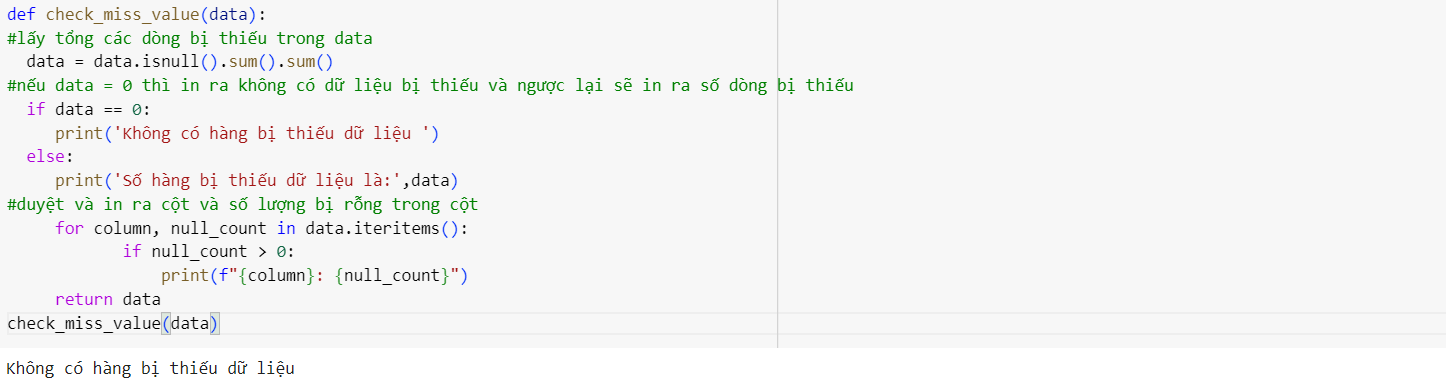
-Bỏ các cột không liên quan



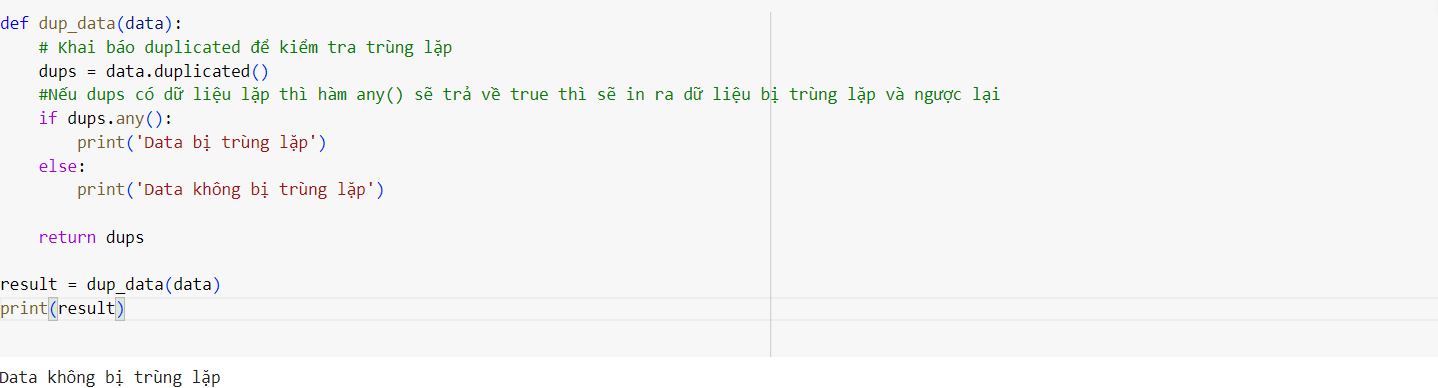
***Các cột ta sẽ lấy trong bài phân tích là :***

1. 'Days for shipping (real)' - Số ngày vận chuyển (thực tế)
2. 'Days for shipment (scheduled)' - Số ngày vận chuyển (dự kiến)
3. 'Customer Country' - Quốc gia của khách hàng
4. 'Sales per customer' - Doanh số bán hàng trên mỗi khách hàng
5. 'Delivery Status' - Tình trạng giao hàng
6. 'Late\_delivery\_risk' - Rủi ro giao hàng trễ
7. 'Customer City' - Thành phố của khách hàng
8. 'Customer Segment' - Phân khúc khách hàng
9. 'Sales' - Doanh số bán hàng
10. 'Shipping Mode' - Phương thức vận chuyển
11. 'Product Card Id' - ID của sản phẩm
12. 'Product Category Id' - ID của danh mục sản phẩm
13. 'Product Name' - Tên sản phẩm
14. 'Product Price' - Giá sản phẩm
15. 'Market' - Thị trường
16. 'Product Status' - Tình trạng sản phẩm
17. 'shipping date (DateOrders)' - Ngày vận chuyển (DateOrders)
18. ‘Type’ - Kiểu

-Xóa các hàng bị thiếu



-Kiểm tra giá trị trùng lặp



-Tạo đặc trưng mới

Tập dữ liệu chứa cột shipping date (DateOrders) là đối tượng DateTime mà từ đó chúng ta có thể trích xuất Tháng, Năm, Ngày và Ngày trong tuần

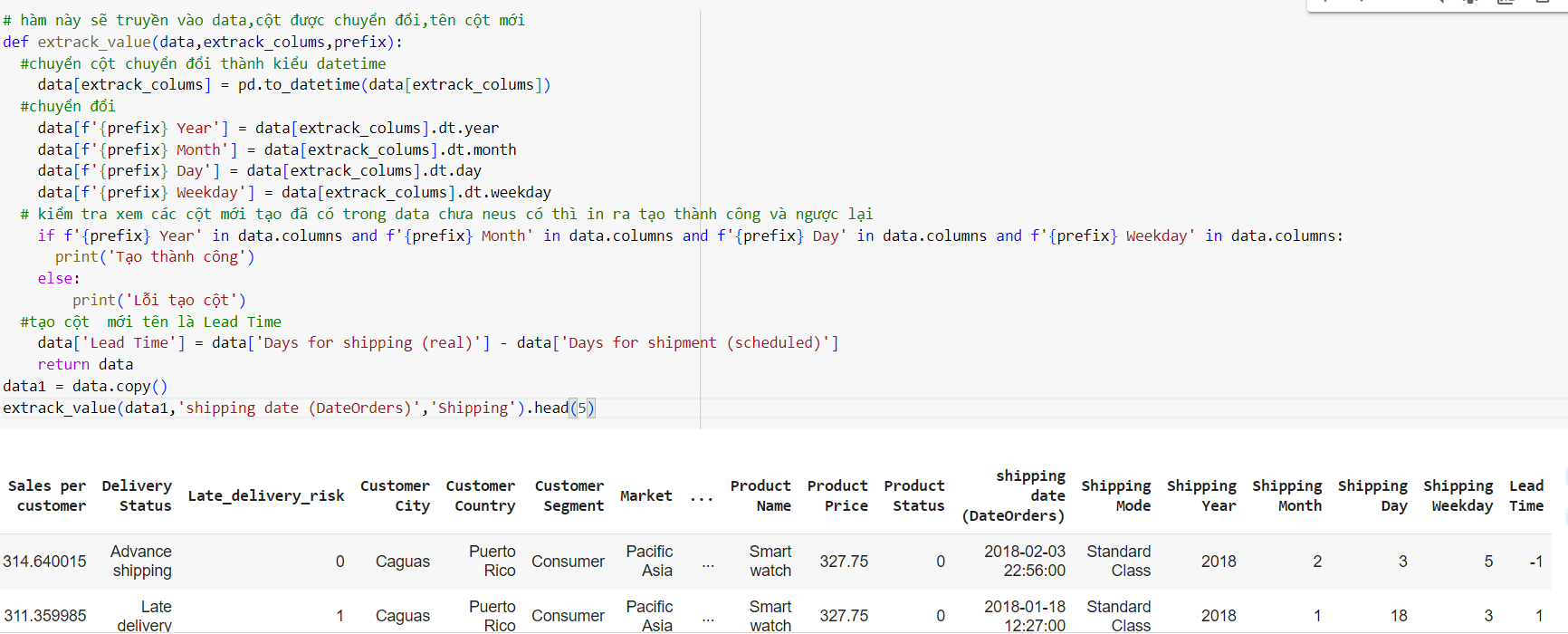
Tháng - để nắm bắt số tháng cho mỗi lần bán hàng.

Năm - để nắm bắt số năm trên mỗi lần bán hàng.

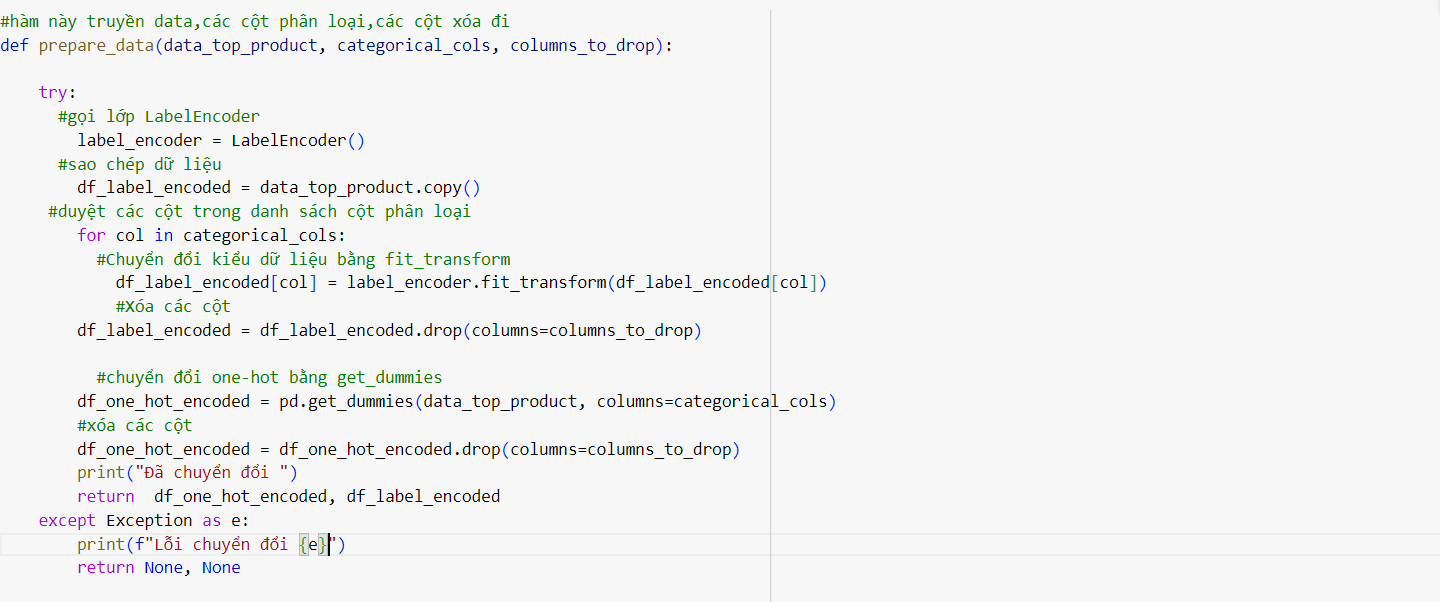
Ngày - để ghi lại ngày trên mỗi lần bán hàng.

Ngày trong tuần - để ghi lại ngày trong tuần cho mỗi lần bán hàng.

Lead\_time, đây là sự khác biệt giữa Số ngày vận chuyển (theo lịch trình) và Số ngày vận chuyển (thực).Điều này sẽ giúp ta hiểu được tác động của thời gian thực hiện đối với nhu cầu.



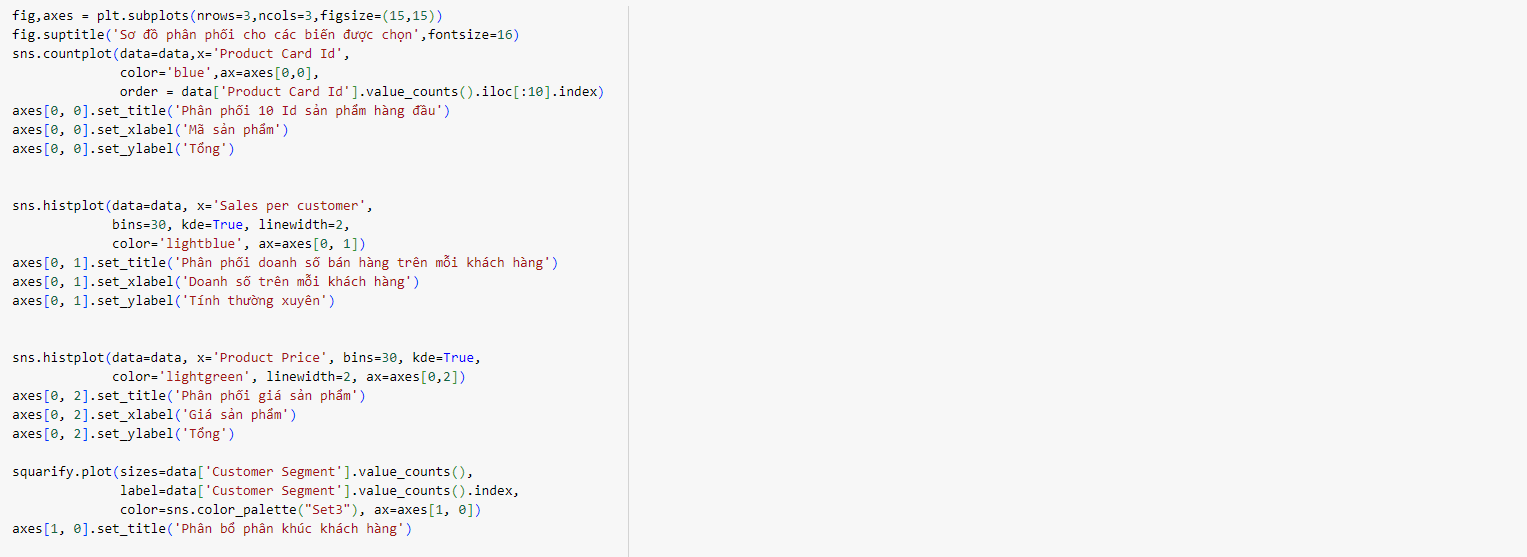
-Chuyển đổi đặc trưng phân loại thành số

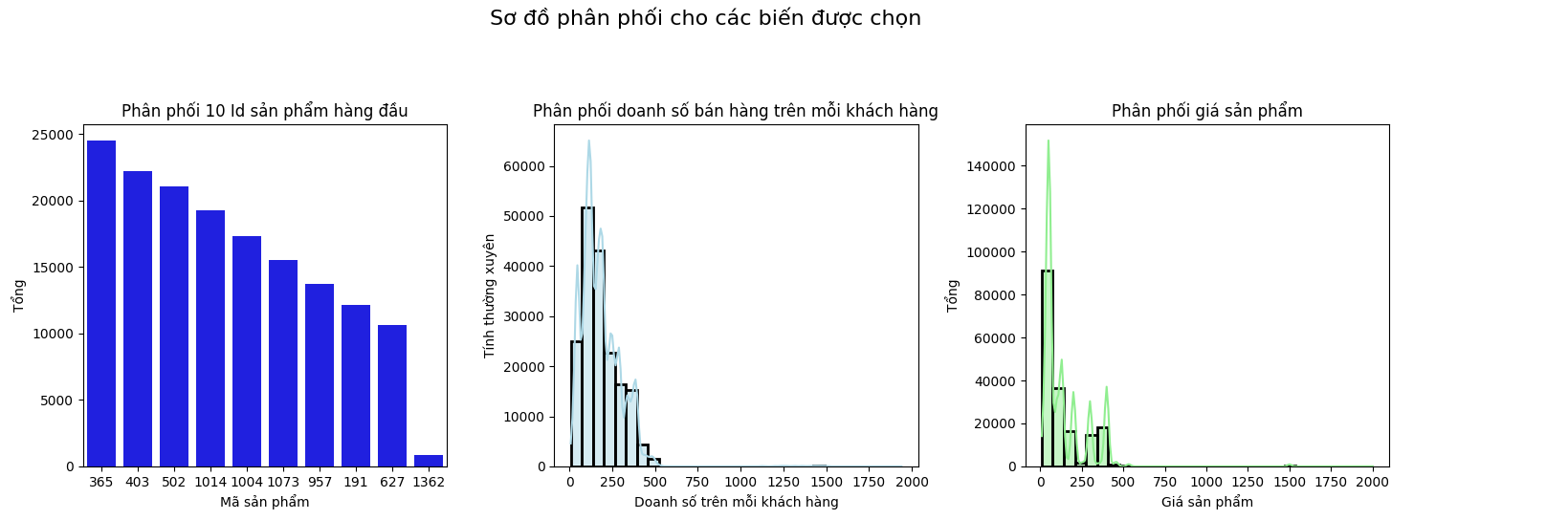




### 7.2.3 Trực quan hóa

-Phân tích đơn biến





Theo các biểu đồ trên ta thấy được:

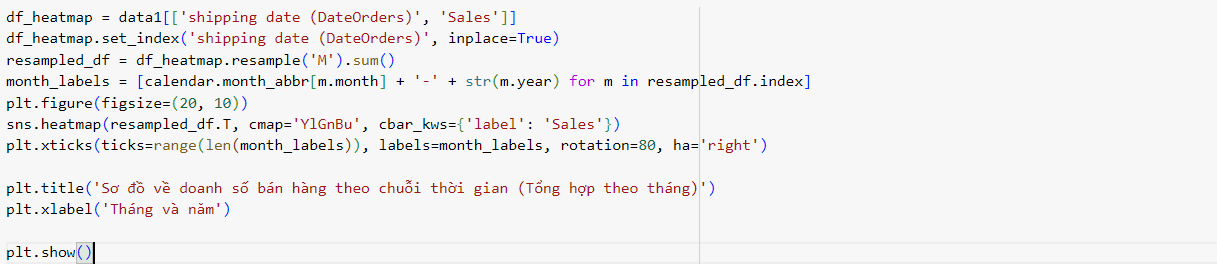
ID sản phẩm bán chạy nhất là 365 tương ứng với tên sản phẩm: Perfect Fitness Perfect Rip Deck.Nhóm sẽ tập trung quá trình dự báo nhu cầu vào sản phẩm này trong tương lai

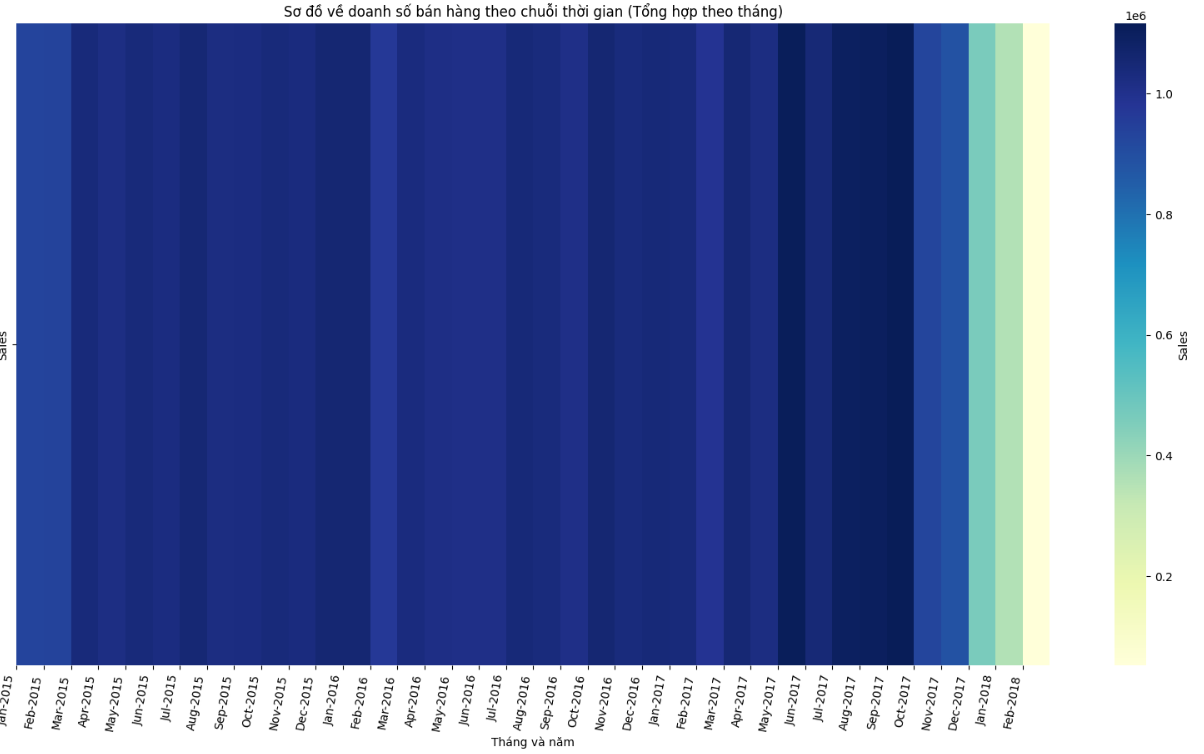
Sự phân bổ giá trị bán hàng và doanh số bán hàng trên mỗi khách hàng đều có độ lệch dương với một cái đuôi dài. Điều này cho thấy phần lớn doanh số bán hàng là dành cho các sản phẩm có giá trị thấp. Đây là một thông tin chi tiết thú vị vì nó có thể gợi ý rằng phần lớn khách hàng rất nhạy cảm về giá.

Phân phối giá sản phẩm cũng có độ lệch dương với đuôi dài. Điều này có nghĩa là phần lớn sản phẩm là sản phẩm có giá trị thấp.

Sự phân bổ của phân khúc khách hàng cho thấy phần lớn khách hàng đều thuộc phân khúc người tiêu dùng.

-Phân tích chuỗi thời gian thăm dò





Theo biểu đồ trên ta thấy được:

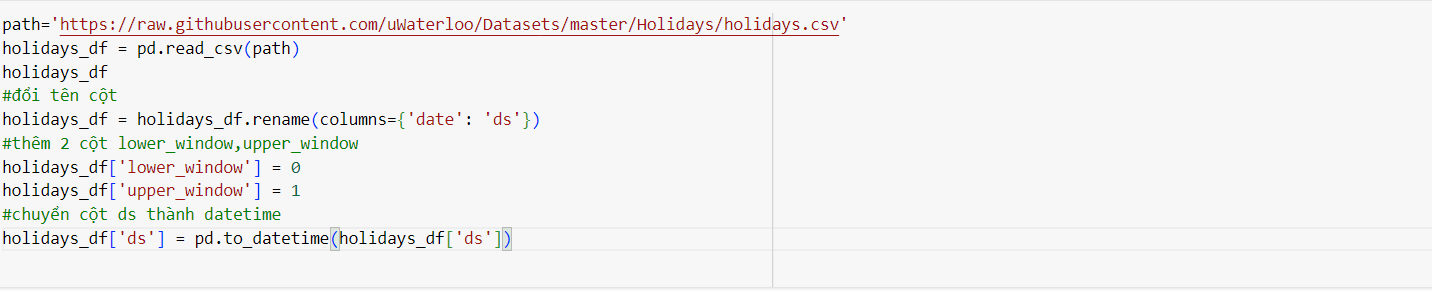
Đánh giá tính nhất quán về sắc thái của bản đồ nhiệt, chúng ta có thể thấy rằng nhu cầu về sản phẩm bán chạy nhất khá ổn định theo thời gian. Tuy nhiên, điều thú vị cần lưu ý là số lượng doanh số bán hàng được ghi nhận trong quý đầu tiên của năm 2015, 2016 và 2017 vẫn ổn định tuy nhiên trong năm 2018 số lượng doanh số bán hàng được ghi nhận trong quý đầu tiên đã giảm đáng kể.

### 7.2.4 Xây dựng mô hình

Ở bài tiểu luận này nhóm chúng em tập trung phân tích cho mặt hàng bán chạy nhất của công ty.Chúng em sẽ Sử dụng thư viện Prophet để lập mô hình nhu cầu về sản phẩm bán chạy nhất

Các ngày lễ ở Puerto Rico lại được đưa vào mô hình.Bởi vì từ phân tích đơn biến được thực hiện trước đó, ta phát hiện ra rằng hầu hết các đơn đặt hàng đều đến từ Puerto Rico.

-Xử lý dữ liệu ngày lễ



Các yếu tố ngày lễ có thể tác động lớn đến doanh thu nhằm tăng khả năng dự đoán chính xác chúng ta cần thêm yếu tố ngày lễ vào để mô hình dự báo chính xác hơn

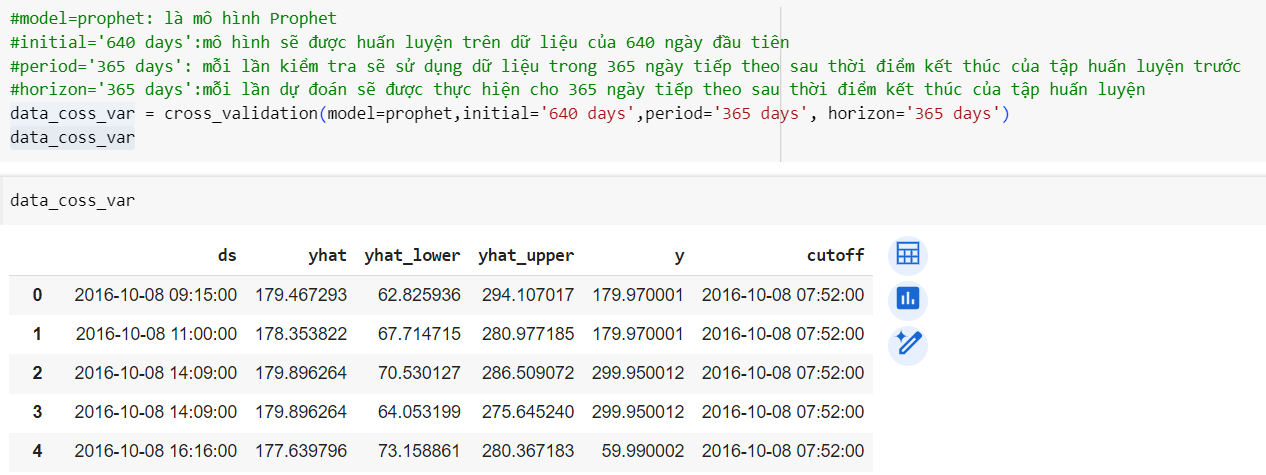
-Xây dựng mô hình



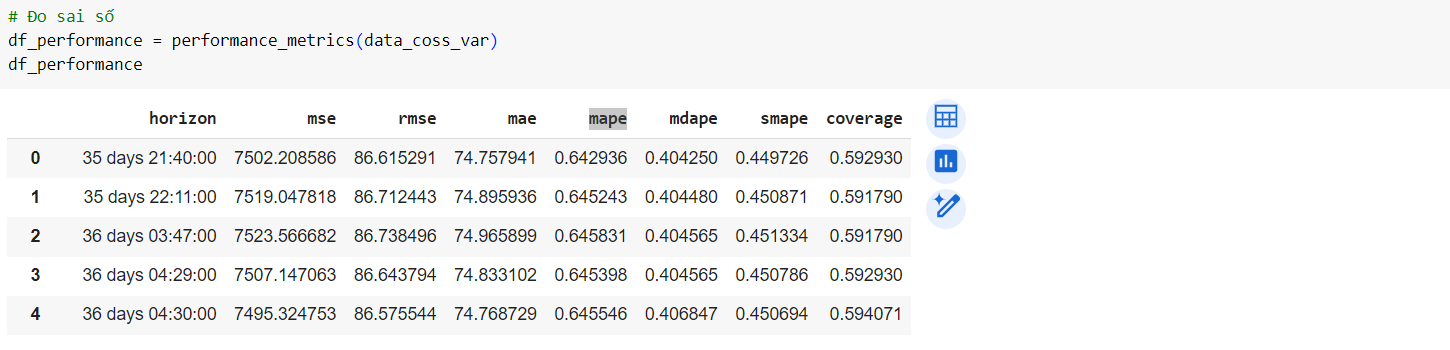
Đoạn mã trên sử dụng thư viện Prophet để mô hình hóa nhu cầu về sản phẩm bán chạy nhất. Mô hình được đào tạo trên cột bán hàng và cột ngày giao hàng. Sau đó, mô hình này được sử dụng để dự báo nhu cầu về sản phẩm bán chạy nhất trong 365 ngày tới

Quy tắc này cũng bao gồm các ngày lễ để tính đến tác động của các ngày lễ đối với nhu cầu về sản phẩm bán chạy nhất. Điều này rất quan trọng vì những ngày nghỉ lễ có thể có tác động đáng kể đến mô hình nhu cầu.

-Đánh giá hiệu suất mô hình

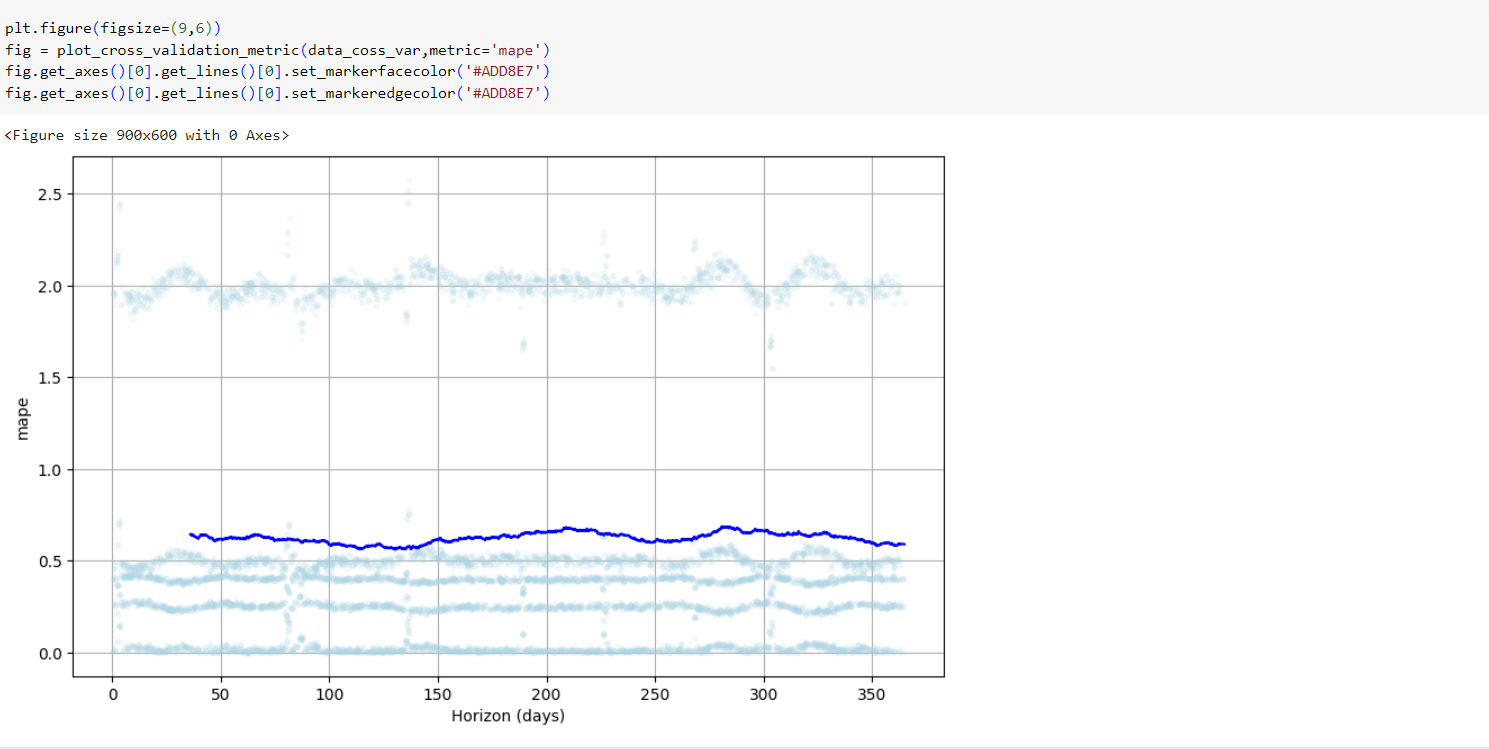


-Độ sai số



Ở đây ta thấy độ sai số của mô hình ngày càng giảm theo thời gian

-Trực quan hóa hiệu suất của mô hình

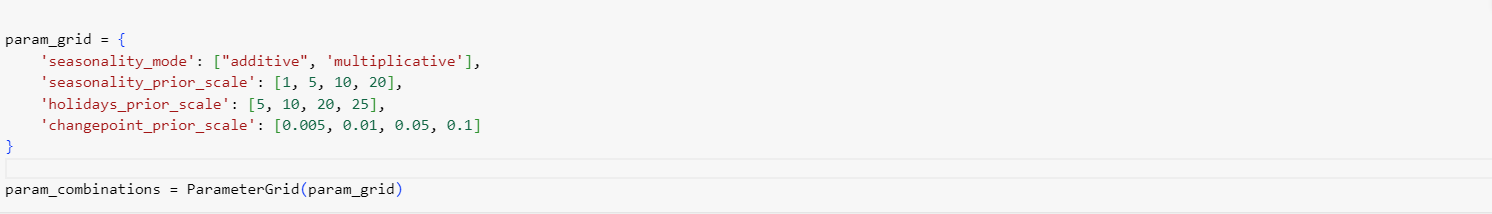


Ở đây ta thấy sai số sấp sỉ khoảng 6% đến 7%

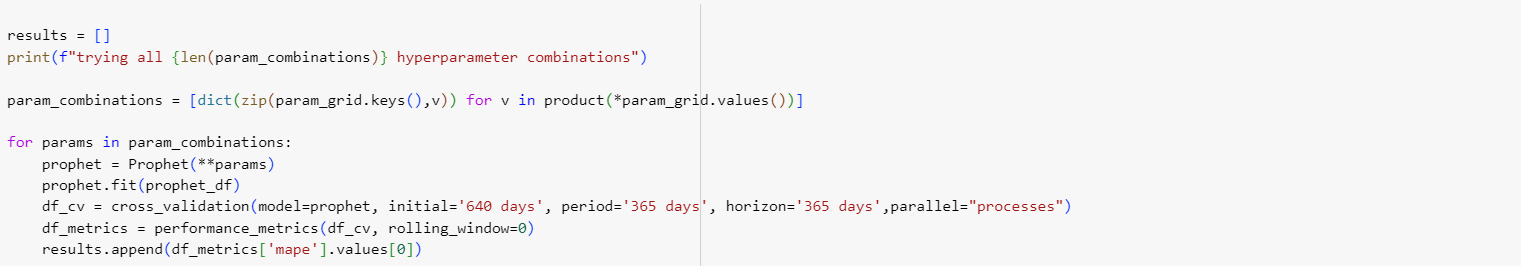
### 7.2.5 Tinh chỉnh mô hình

Sau khi train mô hình ta nhận thấy ta cần tìm kiếm các tham số để tối ưu sai số cho mô hình,chúng ta hãy thử đi tìm và xem mô hình có thể cải thiện được bao nhiêu % sai số

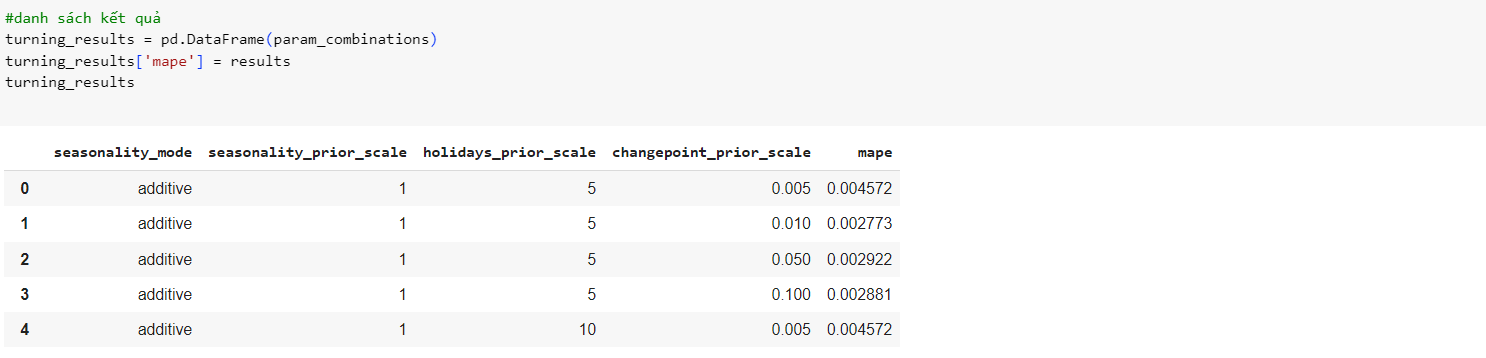
-Tạo siêu tham số

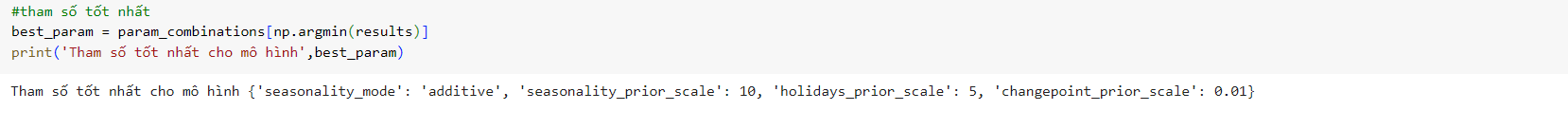


-Training mô hình với các siêu tham số

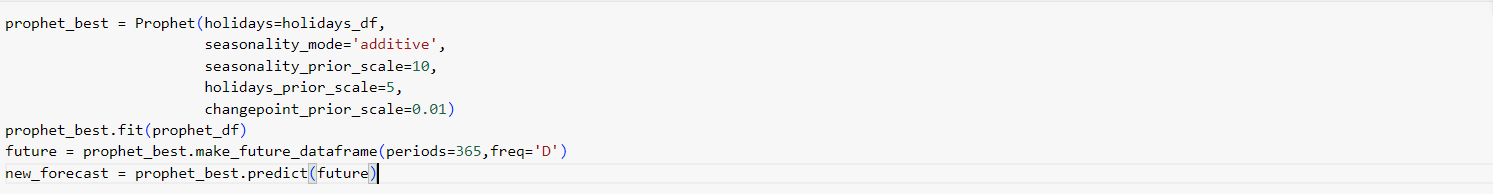


-Kết quả

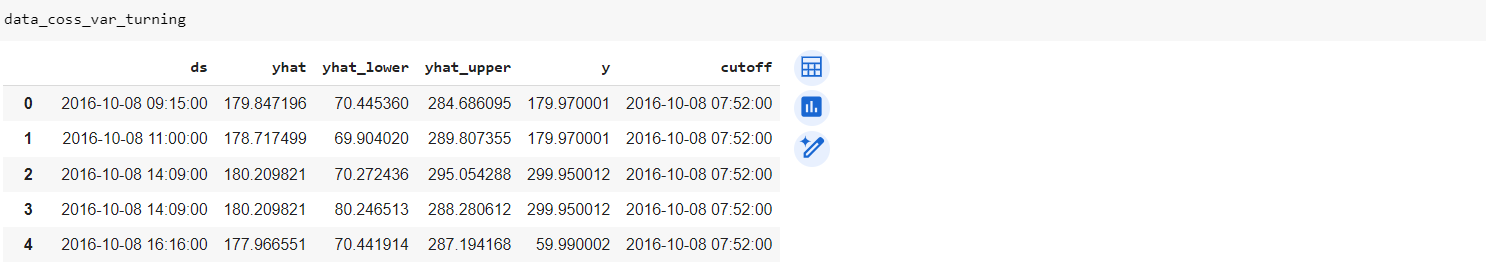




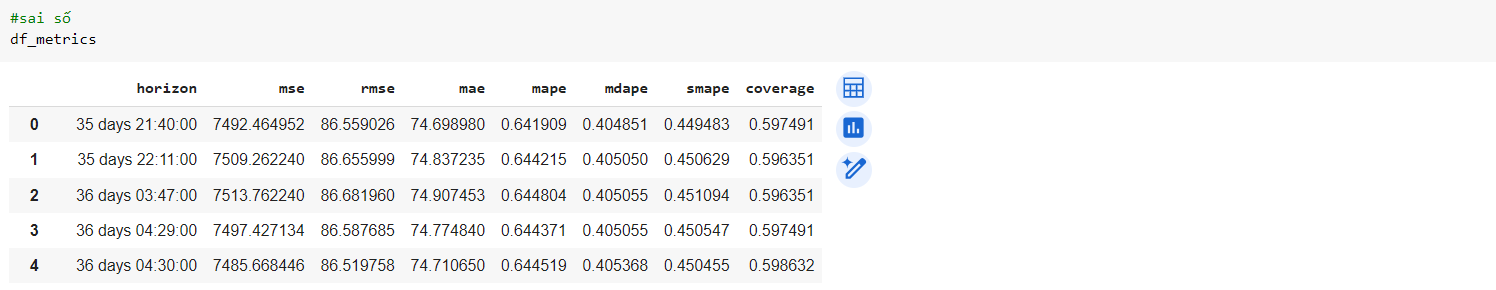
-Xây dựng mô hình với tham số mới tinh chỉnh



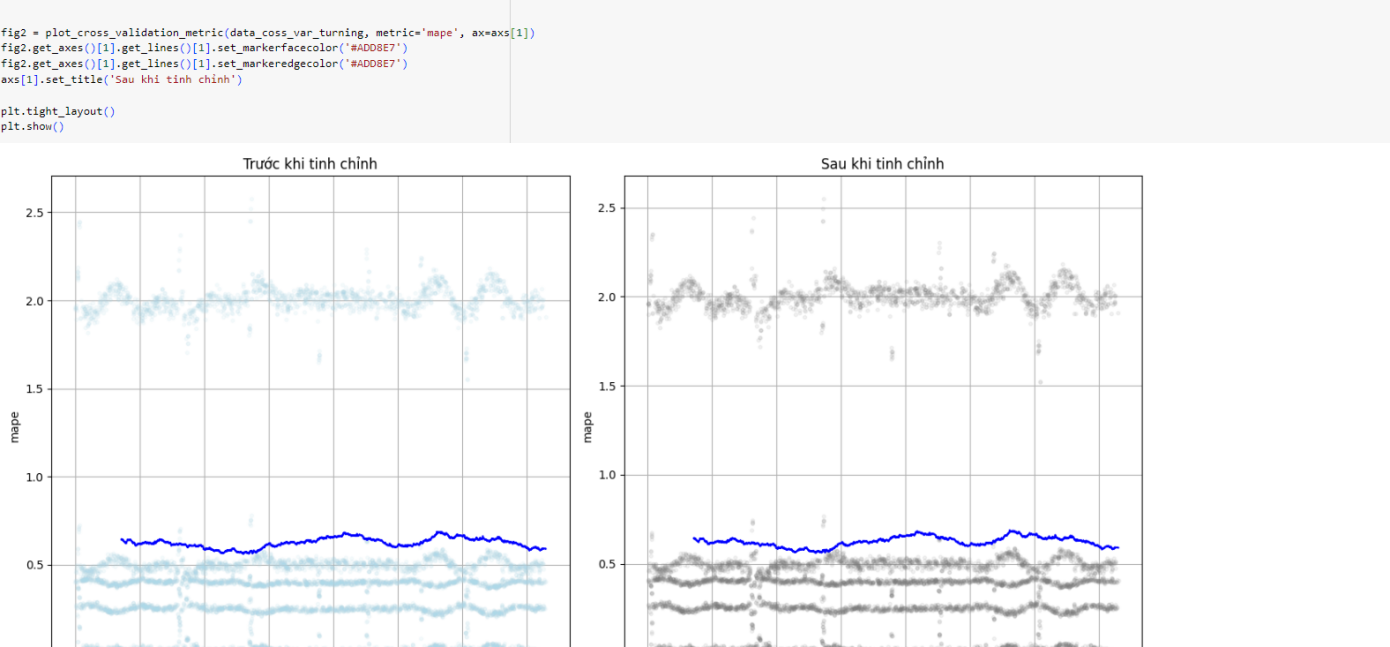
-Đánh giá mô hình



-Sai số



-Trực quan hóa



=>Ta thấy sai số của mô hình được cải thiện,nhưng cải thiện với một tham số rất là bé

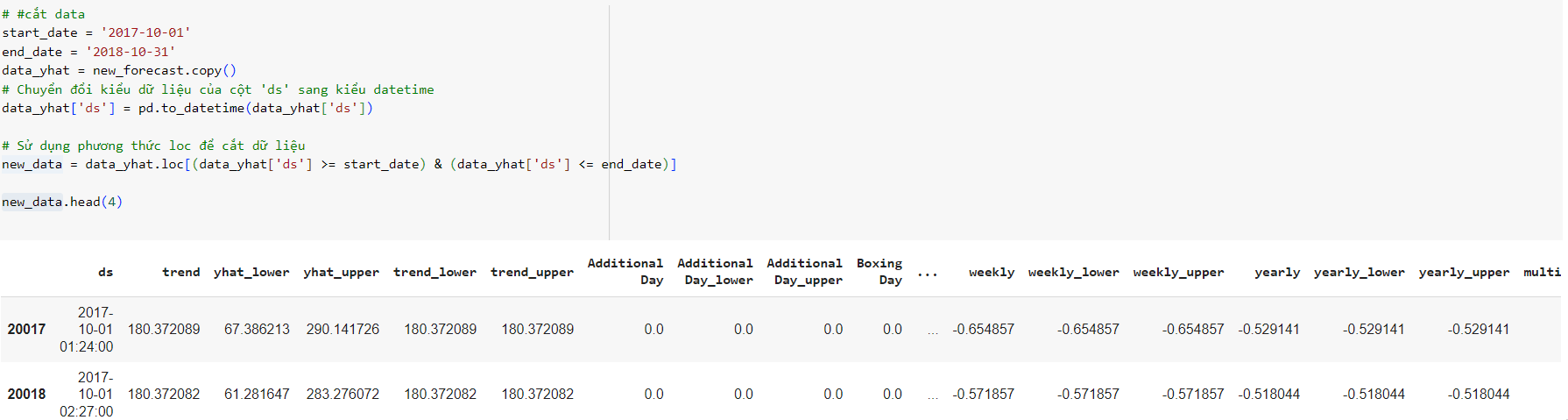
-Biểu đồ dự đoán



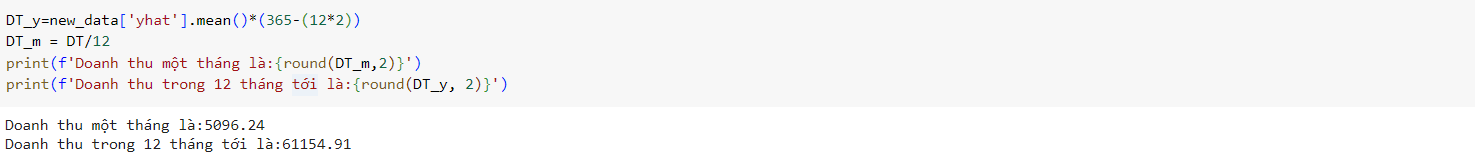
Ở biểu đồ này ta thấy mô hình đã dự đoán từ tháng 10-2017 đến tháng 10-2018.Và ta sẽ tiến hành trả vấn đề thứ nhất doanh thu trong 12 tháng tới là bao nhiêu?

**Ở vấn đề thứ nhất:**

-Cắt dữ liệu dự đoán được trong 12 tháng



-Tính doanh thu của sản phẩm bán chạy nhất trong 12 tháng tới



Em sẽ tính doanh thu trong 12 tháng tới bằng cách lấy trung bình doanh thu một ngày nhân cho 365 ngày tới (trừ đi các ngày nghĩ) ở đây sẽ có 2 ngày nghĩ thứ 7 và chủ nhật

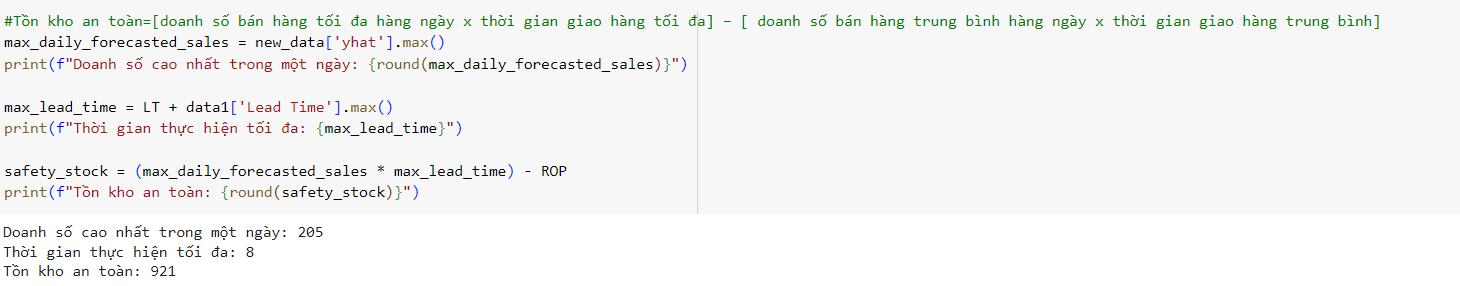
**=>** Kết quả là doanh thu trong 12 tháng tới là 611,154,91 đô

### 7.2.6 Tính toán mức đặt hàng lại,mức tồn kho an toàn,mức đặt hàng

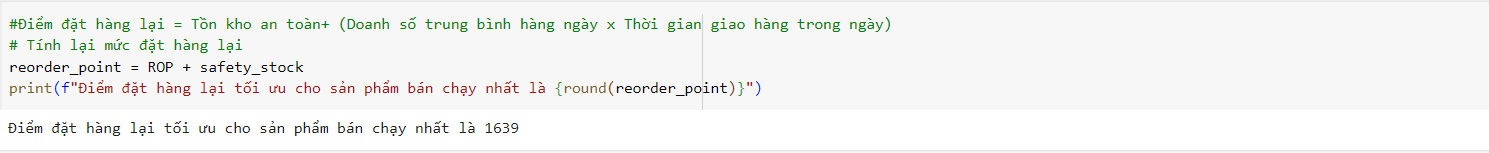
***-***Điểm đặt hàng lại là 717



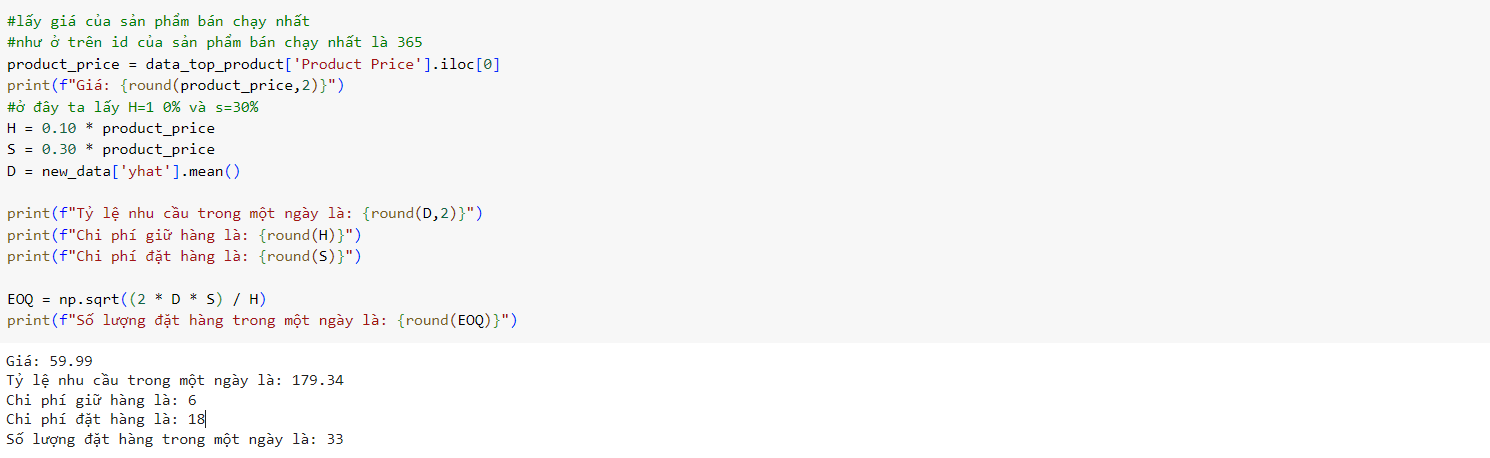
-Mức tồn kho an toàn là 921



***-***Điểm đặt hàng lại tối ưu là 1639



-Số lượng đặt hàng(EOQ) là 33



Sau khi tính toán thì ta có thể trả lời cho vấn đề thứ hai

**Ở vấn đề thứ hai:**

Chúng ta nên dự trữ tồn kho an toàn bao nhiêu là nếu có biến cố sẩy ra? Ở mức tồn kho như thế nào thì chúng ta có thể đặt hàng? Đặt bao nhiêu để tối ưu hoá chi phí?

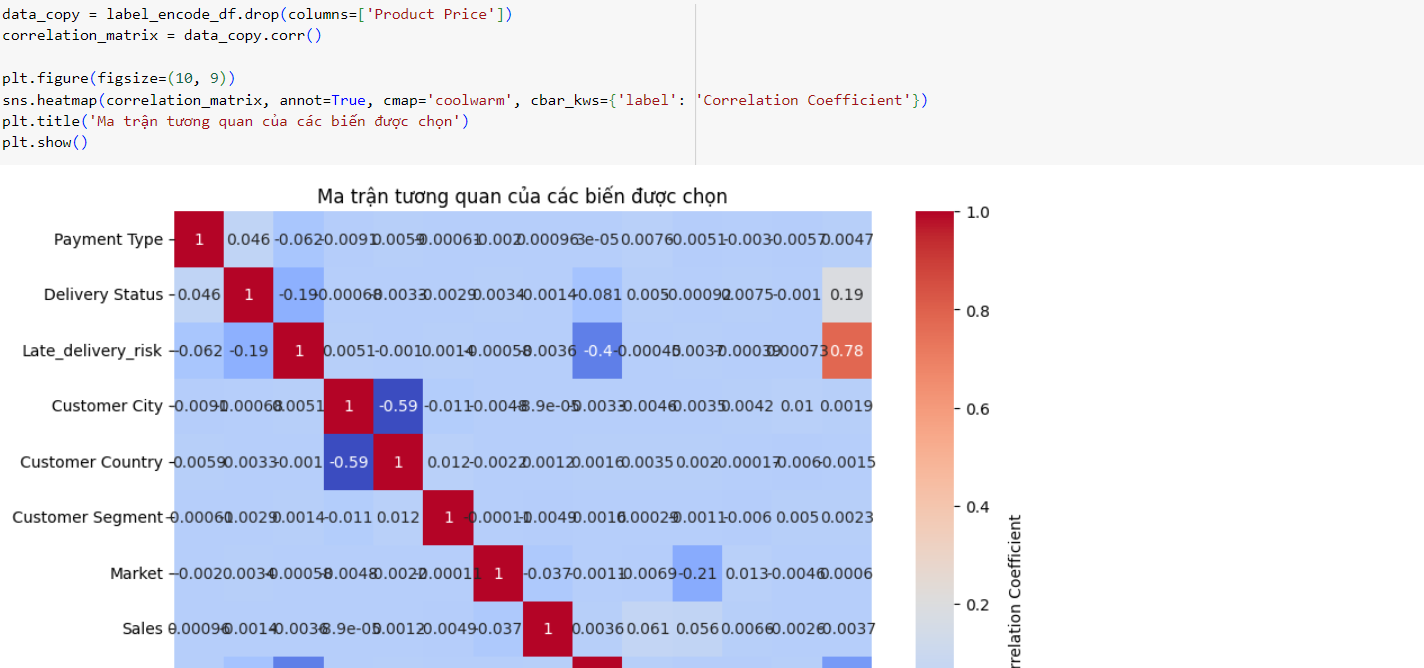
**=>**Chúng ta nên dự trữ tồn kho an toàn khoảng 921 đơn vị,khi mức tồn kho xuống dưới 1639 đơn vị thì ta có thể đặt lại hàng và mỗi lần đặt là 33 đơn vị

### 7.2.7 Các yếu tố ảnh hưởng đến kết quả dự đoán

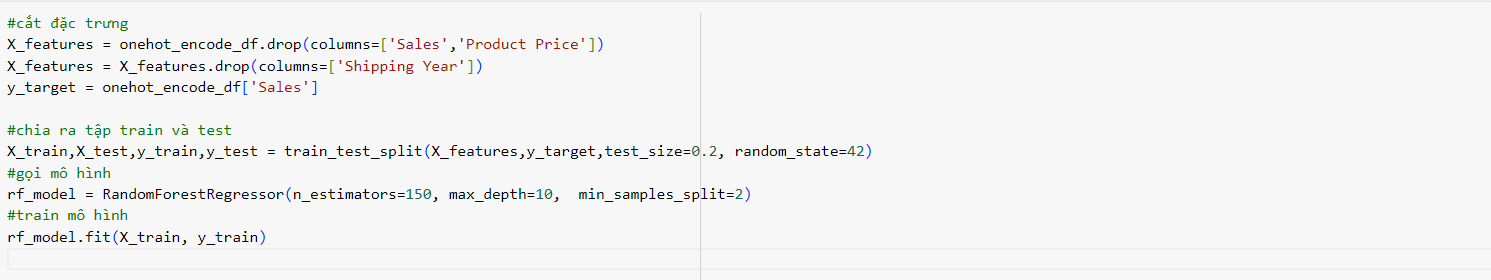
Nhóm em nghĩ rằng các yếu tố như thời gian giao hàng,phân khúc khách hàng và một số yếu tố địa lý.Bởi vì giá sản phẩm luôn cố định nên em nghĩ là sẽ không ảnh hưởng gì tới dự đoán

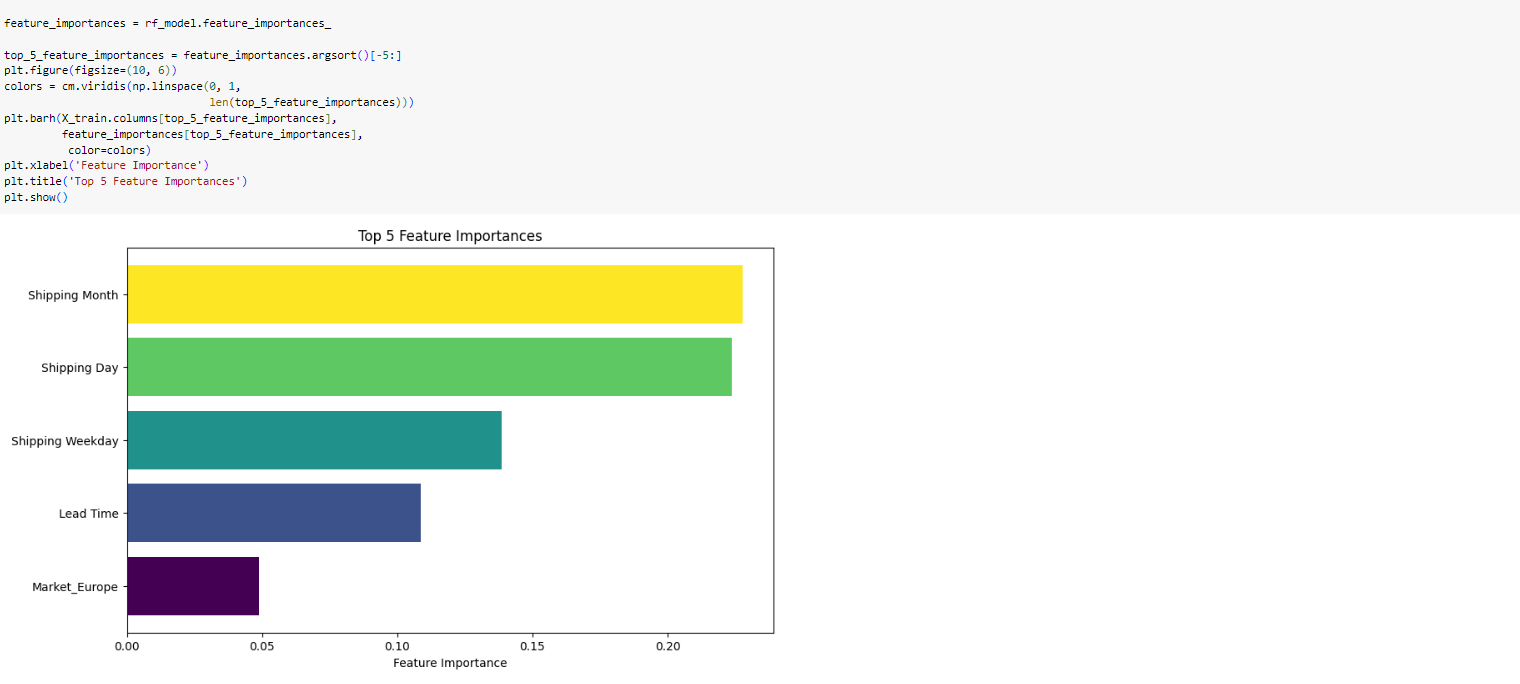
Chúng em sẽ sử dụng thuật toán RandomForestRegressor và thuộc tính feature\_importances\_ để xác định top 5 đặc trưng ảnh hưởng đến dự báo

-Ma trận tương quan các biến được chọn



-Xây dựng mô hình



-Trực quan các đặc trưng ảnh hưởng 

**=>** Ở đây ta thấy các yếu tố ảnh hưởng đến dự báo đó chính là thời gian em có một giả thuyết rằng các ngày lễ có ảnh hưởng rất lớn đến dự báo của mô hình và một phần nhỏ là bị tác động bởi thị trường tiêu thụ

**=>**[**Link Demo**](https://colab.research.google.com/drive/1UWIv_VWV6HiGLWCiz5HjuTjHqwNNtI0B#scrollTo=AUUgRWkQwdC_)

# TỔNG KẾT

## 8.1 Tóm tắt nội dung và kết luận

- Trong tiểu luận này, chúng ta đã sử dụng mô hình dự báo Prophet để dự đoán doanh thu và thực hiện quản lý tồn kho trong một kịch bản thương mại điện tử. Sự kết hợp giữa dự báo doanh thu và các chiến lược quản lý tồn kho là một phần quan trọng trong việc đảm bảo sự cân bằng giữa cung và cầu, đồng thời tối ưu hóa hiệu quả hoạt động kinh doanh.

* Dự báo doanh thu: Sử dụng mô hình Prophet của Facebook để dự đoán nhu cầu trong tương lai dựa trên dữ liệu lịch sử về doanh thu.
* Tính toán điểm đặt hàng lại (ROP): ROP là mức tồn kho mà khi xuống đến mức này, doanh nghiệp cần đặt hàng mới để đảm bảo không xảy ra tình trạng thiếu hụt hàng hóa.
* Tính toán kho an toàn : là lượng hàng tồn kho dự trữ để bù đắp cho những biến động không lường trước được trong nhu cầu hoặc nguồn cung.
* Tính toán số lượng đặt hàng lại EOQ: là số lượng sản phẩm mà doanh nghiệp cần đặt hàng mỗi lần.

-Xác định các đặc trưng ảnh hưởng đến dự báo: Sử dụng thuật toán Random Forest để xác định các đặc trưng trong dữ liệu có ảnh hưởng đến dự báo doanh thu. Random Forest là thuật toán học máy có khả năng học hỏi mối tương quan phi tuyến tính giữa các biến, từ đó giúp xác định các đặc trưng quan trọng một cách hiệu quả.

-Dự báo doanh thu và quản lý hàng tồn kho là một vấn đề quan trọng đối với các doanh nghiệp. Mô hình dự báo doanh thu bằng Prophet kết hợp với ROP, tồn kho an toàn và EOQ là một công cụ hữu ích giúp doanh nghiệp giải quyết vấn đề này một cách hiệu quả.

## 8.2 Kết quả đạt được và điểm còn hạn chế

-Kết quả đạt được:

Dự báo được doanh thu của doanh nghiệp trong 12 tháng tới

Tính toán được mức đặt hàng lại,mức tồn kho an toàn và số lượng đặt hàng khi đặt hàng lại lại

Xác định được các đặc trưng ảnh hưởng đến dự báo

-Điểm còn hạn chế:

# Mô hình còn sai số cần được cải thiện hơn

Đòi hỏi dữ liệu lịch sử đầy đủ, chính xác và có nhiều đặc trưng.

Thuật toán Random Forest có thể gặp khó khăn trong việc giải thích kết quả.

# DANH MỤC TÀI LIỆU THAM KHẢO

| [1] | Jason Brownlee- “[Data Preparation for Machine Learning](https://classroom.google.com/c/NjQ5MDAzODc1MDI5/a/Njc4MTcyMjUyNjU1/details)” (2020). |
| --- | --- |
| [2] | Datacamp-“[Random Forest Classification with Scikit-Learn](https://www.datacamp.com/tutorial/random-forests-classifier-python)” (01/05/2024) |
| [3] | Scikit learn -“[sklearn.ensemble.RandomForestClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)” (01/05/2024) |
| [4] | Scikit learn -“[sklearn.model\_selection.ParameterGrid](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterGrid.html)” (01/05/2024) |
| [5] | Tutorials-”[Label Encoding in Python](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/)” (01/05/2024) |
| [6] | Phamdinhkhanh-“[Ý tưởng của mô hình rừng cây](https://phamdinhkhanh.github.io/deepai-book/ch_ml/RandomForest.html)” (01/05/2024) |
| [7] | ICHI.PRO-“[Phân tích chuỗi thời gian với Facebook Prophet: Cách thức hoạt động và Cách sử dụng nó](https://ichi.pro/vi/phan-tich-chuoi-thoi-gian-voi-facebook-prophet-cach-thuc-hoat-dong-va-cach-su-dung-no-265389504990778)” (01/05/2024) |
| [8] | TabML-”[Mã hóa one-hot](https://machinelearningcoban.com/tabml_book/ch_data_processing/onehot.html)” (01/05/2024) |
| [9] | NguyenTienSu-”[DP4ML - Feature Selection - Phần 6 - How to Use Feature Importance](https://tiensu.github.io/blog/91_data_prepeation_for_ml_feature_selection_6/)” (01/05/2024) |
| [10] | [tcuongd](https://github.com/facebook/prophet/commits?author=tcuongd) -“[diagnostics](https://facebook.github.io/prophet/docs/diagnostics.html)” (01/05/2024) |
| [11] | [tcuongd](https://github.com/facebook/prophet/commits?author=tcuongd) -“[facebookprophet](https://github.com/facebook/prophet)” (01/05/2024) |
| [12] | [tcuongd](https://github.com/facebook/prophet/commits?author=tcuongd) -“[PROPHET](https://facebook.github.io/prophet/docs/quick_start.html)” (01/05/2024) |
| [13] | [Chat GPT](https://chatgpt.com/) |
| [14] | [Gemini](https://gemini.google.com/) |